JPRS-UCC-85-003

23 April 1985

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service (NTIS), Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semimonthly by the NTIS, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

JPRS-UCC-85-003

23 April 1985

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

CONTENTS

GENERAL

HARDWARE

SOFTWARE

APPLICATIONS

THEORY OF COMPUTATIONS

PUBLICATIONS

GENERAL

COMPUTER-AIDED INSTRUCTION GAINS, SETBACKS

Moscow KOMSOMOL'SKAYA PRAVDA in Russian 22 Nov 84 p 2

[Article by S. Kushnerev, KOMSOMOL'SKAYA PRAVDA special correspondent, Odessa: "VUZ's: Forecast for Tomorrow; Computer Poses Questions; Computerization of Education--Fad or Economic Necessity?"]

[Text] The first sign which portended that the hour of the universal computerization of education had struck proved to be the totally unsuspecting tutor. After all, tutors appeared precisely when there began to be not enough teachers. Or is it that there are no longer enough teachers for everyone?

Socrates, at one time renowned for his leisurely discussions with students, probably would be dismissed from today's secondary school for professional incompetence, for not having fit into the program. In a class in which a 45-minute lesson is set aside for 30 students, the teacher more and more often is forced to act automatically. And the capabilities of automated systems are at the same time approaching human. And, then, once upon a time...

"Hello! I am the 'Agat' personal computer. I will help you master the skills of speed typing with 10 fingers blindfolded," lit up on the screen of a terminal.

Aha! I was consumed with almost athletic fervor. The fact is that once they had already tried to teach me to type. And rather unsuccessfully--to this day I type "incorrectly," with one finger. And so, I said to myself, not without malicious pleasure, "Typing--that's appropriate." As they say, this lesson will be something for me to stand up to.

However, my new "teacher," externally resembling a video game, quickly made herself be treated with respect. She asked me to "write" on the display's keys any 10 words and she determined my starting level--and it began!

The computer managed to reveal my weakest points and accordingly slipped in the most undesirable exercises. Having made a blunder with the letter "ts" [in Cyrillic], I then and there had to knock out "Tsygan na tsypochkakh skazal tsyplenku: tsyts!" [The gypsy on tiptoes said to the chicken, "Shhh!"]. She was absorbed with me alone--this is nice in a lesson, too.

But I had hardly stopped taking the test when "tsypochki" [tiptoes] and "tsyplyata" [chickens] were replaced with the stipulations of the SAN ("general physical and mental state, activity, mood") test and of the (Lyusher) color test, After a minute checkup, all on the same screen, a diagrammatic portrayal of me cropped up and ordered me with musical accompaniment to repeat after it in synchronism autogenous training exercises. The organism itself, without a bell, assigned itself a "break."

To make a long story short, if just the desire to master speed typing had brought me to Odessa to the "Problems of Computer-Aided Instruction for Users of Organizational Management Systems" scientific seminar and conference, I think after a 5-day business trip something happened for which an entire semester would not have sufficed at the university.

This is not only what I think--the data of special studies show that whereas 400 hours are required for the average student to reach a speed of 180 characters per minute when teaching by the traditional method, communication with a computer accelerates this process 8-fold! So, if desired, it would have been possible to pack up in five days.

As a second lesson I could choose "French," "Russian as a Foreign Language," or any other language, including machine languages. By replacing a cassette with a program, I could study civil law, the theoretical fundamentals of radio engineering and formal logic--dozens of these instruction programs have already been created. I could even, having devised some kind of science of my own, learned to write it in the form of a computer-aided course, according to which the same computer could teach this science of mine to others.

But let he who is sure of his pedagogical abilities put together the best of these courses. After all, the machine does not in actual fact teach! It is a human being who puts into it, together with a certain sum of knowledge, the powers of observation, keenness, resoluteness, vigilance and "carrot and stick" of pedagogy. It is no wonder that not only the best specialists in the respective fields, but also the best teachers, psychologists and specialists in the methodology of teaching, are enlisted today for putting together courses for computer-aided instruction systems.

I remember, in our school the Russian Language and Literature teacher had a reputation of being quite a character. But once he suddenly "dashed one off" which we had already heard. Then, once again, and again. Finally, we understood: After long years of working on the school "assembly line" he simply began to forget how he had already managed to joke in the class at hand.

How simpler it would have been to circulate his best lesson, even the quintessence of best lessons, and to update it only as necessary. If a teacher does not have every day to confirm already scored successes, there will be more time left for him for new ones.

A computer is not a teacher's competitor, just as a sewing machine is not a competitor of a tailor, a combine of a farmer and a tape recorder or record player of a musician.

"Repetition is the mother of knowledge?" Is it not unfortunate to force an entire army of intelligent and educated people to repeat every day?

Of course, a copy is always worse than the original. But, let us consider this: An individual tutor at State expense is too great a luxury. At one's own expense, too. However, it is precisely self-training which is occupying more and more time in humanity's educational timetable. The higher the level of education, the stronger the bent for self-education. And who of us does not know from his own experience that in self-education especially acutely missing is some kind of teacher ready at any moment to test, encourage, insist, prompt and, finally, intrigue!

Having a personal computer, which, they say, in the foreseeable future will appear in retail stores, and a set of programs for it, it will be possible in the most remote corner of the country to "study," let us say, with Professor Losev from his 6-volume "Historiya antichnoy estetiki" [Historiya Aesthetics in Antiquity], or with the best specialists in any other field of knowledge. In the same manner, any one of us, having found a half hour, will be able directly at home to develop skills in speed reading, accuracy in reacting, or even oratorical abilities.

The paradox of the epoch of the NTR [scientific and technical revolution] is that the universal computerization of education is necessary precisely in order to reveal and develop the individual unique abilities of everyone.

And this problem is not simply a scientific one, but a social one.

In order to solve it by the old measures, at least one half of the country's population would have to be studying with a tutor.

They told me about one of the already existing computer-aided courses--the laboratory course "Geological Prospecting Modeling." Its classical version is a box with sand in which a buried sheet of paper simulates the boundary of the occurrence of mineral resources. By piercing the sand with a stick, substituting for a drilling rig, the student determines the structure of the unknown "stratum" and calculates the reserves. Instead of a box with sand, the computer presents to us on the screen a chart and initial data right down to the time of year and the climatic situation. You--"one in a field of soldiers"--act at your own risk as the entire geological team and on the screen is registered: "Did Not Fit into Field Season," "Overexpenditure in System of Underground Routes," and the like. They say that not a single video game engrosses you in this way!

But the first lesson was not accidentally precisely typing. Today this computer-aided course is one of the most widespread. After all, through it not only professional typists are trained, but at the same time also keyboard equipment operators. In other words, this is the first and most elementary step toward communicating with a computer, which day by day will be ever more necessary in quite different fields. All of us--workers, engineers, kolkhoz chairmen, students, schoolboys, and scientists--must as quickly as possible become "control system users," i.e., learn to use computer equipment. And the struggle for this "campaign against illiteracy" is still ahead. And this is already a national economic goal.

It is time to give a realistic portrayal of its scale. Recently a determination was made of the required level of training in the field of computer technology for all VUZ fields of specialization, except (for the time being!) fields of specialization in art. This level is no joke: The amount of training is from 60 to 120 to 300 to 450 hours! Are the teachers themselves ready for this? In the opinion of the chairman of the scientific council for the development of computer-aided teaching systems, the director of the Scientific Research Institute of Problems of Schools of Higher Learning, A.Ya. Savel'yev, elementary "computer literacy," even among them, can be achieved in five to seven years upon condition of the concentration of all forces. Speaking about the concentration of forces, let us recall that only half of VUZ's are under the jurisdiction of the USSR Minvuz [Ministry of Higher and Secondary Specialized Education], and the remainder are "scattered" over branch ministries. And the personnel for the school, vocational and technical school, and technical school, in which school reform has been stipulated for the study of the fundamentals of computer technology?

In some foreign countries the very manufacturers of computers have taken on themselves the training of personnel: In selling a computer to a school or VUZ, the company trains teachers in "computer literacy" free of charge.

A solution to this problem is necessary one way or another. That is, a centralized one.

Incidentally, computer-aided instruction solves the problems not only of yesterday's specialists, but also of today's students.

Let us say, for a long time there have been arguments about the free attendance of lectures. And the VUZ, as they say, is there now. Meanwhile, the "pedagogical service" of the computer has testified to the fact that the speed of the mastery of instructional material even within a seminar group can differ by a factor of 2.5! This figure says that in one auditorium someone will always ask, "Slower, I am not able to write," and someone, having mastered the main point in a snap, will from boredom solve crosswords, knit, or doze.

Computer-aided teaching systems make real what today a VUZ can only dream of. Every student will be able to study on an individual schedule and be able to take any course at any time without attending classes. It will be possible to "spread" a session through a semester and do away with crash studying, and, finally, the numerous problems of night school students and correspondence students can be solved by themselves.

"Yesterday" is driving "today" on. And "today" is balking.

First, in writing a teaching program one hour of machine time costs the teacher dozens of hours of strenuous work. Where is a presentday teacher to get these hours from, and how is he to be paid for them?

Still, of course, there are enthusiasts. And not a few computer-aided courses have already been developed: They do exist in the Unified Bank of Algorithms and Programs. But all the same there is no single editorial office for circulating the best. And the manufacturers are still not especially trained for circulation itself.

Hence, another misfortune. The computers themselves are already present in many VUZ's. But often they do not know what to use them for and, it happens, they almost use them in place of a calculator.

Finally, the main point: Today you do not buy a personal computer in a shop. The equipment already mastered by industry only approximates the necessary parameters, but its production is still insufficient for the needs of education, medicine, the field of servicing, and for the average consumer.

And again I hear about the psychological barrier: They say that people are not psychologically prepared for this. Why, even the automobile in its day would have seemed to be a "growling monster" if it had been judged only through hearsay. The misfortune is that the majority of the population judges minicomputers still precisely in this way.

Someday computers will appear which are not only teaching computers, but which also have broken bad habits, let us say, of resistance to change and reactionism. But for the time being this is not so; we have to count only on ourselves, on our own foresight, which is inaccessible to any machine. Computers are no luxury!

By the way, the first ones to master this were tutors themselves: Some of them, they say, are already using computer-aided teaching systems in their "practice."

8831
CSO: 1863/126

KOMSOMOL ROLE IN PROMOTING COMPUTER LITERACY AMONG YOUTH

Moscow KOMSOMOL'SKAYA PRAVDA in Russian 26 Dec 84 p 2

[Article by N. Sleptsov, head, Department of Scienrific Youth, Komsomol Central Committee: "Dialogue with Computer; All Ages Bow to Computers"]

[Text] In the decree of the CPSU Central Committee titled "On Further Improvement of Party Management of the Komsomol and Increasing Its Role in the Communist Education of Youth" a direct indication is given of the necessity of concentrating the efforts of youth on the basic directions of technical progress, total automation and advances in electronics.

The establishment and development of new trends in modern science and engineering involves the direct participation of Komsomol members and youth, such as in nuclear power and molecular biology, genetic engineering and laser technology. And now there is a new goal, a new highly complex task: electronics.

According to available estimates, as early as the next decade microprocessors will find application in more than 200,000 different kinds of equipment and apparatus for industrial and home purposes. Since the beginning of the present five-year plan period, the production of microprocessors has increased almost 5-fold, and of microcomputers more than 2-fold. About 25,000 industrial robots have been produced.

However, in spite of the scales achieved in the production and introduction of computer equipment, the demand for it is not only not slackening but is constantly growing. Let us consider the figures, which literally stagger the imagination: Next year 57 percent more than at present control computer systems will go into service for automating machines, equipment and instruments! Practically not a single sector of the national economy knows such a development pace--we are witnesses to a kind of revolution.

Electronics is one of the youngest branches of modern science, engineering and production. It is quite natural that more youth are working in it, and they are of a very high educational and professional level.

Youth already today are making a weighty contribution to the development of computer technology. As early as in 1972 the Komsomol Central Committee approved the work of the Armenian, Belorussian, Latvian and Ukrainian republic and Moscow and Leningrad city Komsomol organizations in the enlistment of youth for the introduction into the national economy of computer equipment and automated control systems. The development and use of computer equipment has for many years been one of the main jobs of the Komsomol organization of the Institute of Cybernetics imeni V.M. Glushkov of the Ukrainian SSR Academy of Sciences. Such forms of operation as introduction staffs and posts for continuous Komsomol monitoring of the development of computers of new generations and automated control systems have recommended themselves well here.

A group of young scientists and specialists at the Kazan Computer Plant was awarded the Lenin Komsomol Prize for 1984.

The packages of application programs developed here are considerably increasing the efficiency of the computing process and the saving of labor resources and already at the present time they are being used in more than 1500 computing centers in the country. They are producing a considerable saving.

Attaching exceptionally great importance to the role of the Komsomol and youth in this task, the Komsomol Central Committee, the USSR State Committee on Science and Technology, the Presidium of the Academy of Sciences, Minvuz [Ministry of Higher and Secondary Specialized Education], Minpros [Ministry of Education] and the USSR State Committee on Vocational and Technical Education have made the joint decree titled "On the Participation of Komsomol Members and Youth in Development and Efficient Use of Computer Equipment and in Studying the Principles of Its Utilization." The goals are defined in it and a longterm program of actions is worked out for all detachments of the Lenin Komsomol.

And so, the goals have been defined--but what does their fulfillment begin with? First, of course, it is necessary to reinterpret critically the experience gained. To develop together with the administration a clear program of organizational work and practical actions. It is important to define absolutely from the very beginning the specific directions and the objectives of the sponsorship in which the maximum concentration of the efforts of Komsomol members and youth is required. Each young person must know and understand the role of computing equipment in the development of our society. It is a question first of all not of the direct participation of Komsomol organizations in solving scientific and technical and production problems, but of its political support.

The complexity and scale of the problems reveal a broad scope for the creative quest of young innovators. Especially important here today are new ideas, conceptions and non-traditional solutions. Therefore, it is important to bring in time to a wide range of youth the list of problems requiring a solution. It is necessary to practice the issuing of Komsomol assignments for the implementation of the sponsorship's programs and to include them in personal creative plans. And to take the results of the completion of these assignments into account in summing the results of both an individual and team socialist competition. It is necessary to actively employ the creation of KMK's [expansion unknown] and KMTK's [expansion unknown] in the most important sections for the development and production of new equipment and to organize competitions between them according to the principle of the workers' relay race.

The electronics industry is organically related to quite different branches of modern science and the national economy. And for this reason it is necessary to develop in every way forms of cooperation and to expand direct contacts between the youth of scientific development institutions and manufacturing enterprises. Young specialists, technicians and workers should appear not in the role of simple executors of someone else's ideas and developments, but be coauthors and confederates. And here there is someone to compete with! The "Cybernetics" permanent Komsomol-youth scientific research center has been organized at the Institute of Technical Cybernetics of the Belorussian SSR Academy of Sciences. It coordinates the work of the creative youth associations included in it from quite diverse enterprises and institutions in Belorussia. Over the years of just the current five-year plan period the saving from this combined activity reached more than 1.5 million rubles!

In the opinion of specialists, programming in the future will play for humanity the role of a "second literacy." Beginning with the 1981-82 school year, the "Concept of Programming for the Computer" section was introduced into the program of the algebra course for the eighth classes of secondary schools. In the future each school graduate must have computer operating skills.

In secondary vocational and technical schools and institutions with a period of instruction of more than one year, teaching of the academic subject, the same for all fields, of "Fundamentals of Programming and Computer Technology," is being introduced. The extensive instruction of students in the fundamentals of computer technology and programming, providing three levels of basic and additional training, is being introduced in VUZ's. Plans are being made to establish appropriate educational centers and complexes for the purpose of training broad masses of the population, including working youth.

To attract children and young people to the mastery of equipment is an exceptionally important and crucial task of the Komsomol. A network of cybernetics schools and schools and courses for young programmers has been formed by the combined efforts of scientists and specialists, Komsomol committees and public education agencies. The work of scientists of the Siberian Division of the USSR Academy of Sciences in teaching programming to school children, beginning with the lower grades, has won extensive renown both in our country and abroad. Since 1977 a school information science group has been in operation here, headed by a prominent specialist and immense enthusiast of this task, USSR Academy of Sciences Corresponding Member A.P. Yershov. The all-Union correspondence school for young programmers with its branches already includes thousands of schoolchildren.

Interesting experience has been gained in the schools for young programmers of the "Iskatel'" [Seeker] little academy of sciences in Krymskaya Oblast, and the Institute of Mathematics and Cybernetics of the Lithuanian SSR Academy of Sciences. Thus, various schools have been formed for teaching computers to children, and these schools differ in the teaching method, organization of the teaching process and the programming language studied. And today, with the change to the mass instruction of children and young people in computers it is impossible not to take into account this experience--both positive and negative.

But, in unfolding this work it is necessary from the very beginning to establish a reliable barrier for formalism and working in spurts. Real acts are necessary, and not slogans and reports. The most important thing today is to create the necessary conditions for a general "campaign against computer illiteracy." It is necessary, for example, to reach the point that all machine time which is free from the principal work is used for working with youth. Every city and rayon must have a longterm program for teaching youth the fundamentals of computer technology and programming. Without such a program it is hardly possible to speak of the general coordination of efforts. Obviously, it is necessary to introduce specific changes into the work programs of little academies of sciences and scientific societies of students. It must become a rule that each of their graduates without fail has mastered skills in the use of computer equipment. It is necessary to increase considerably the role of the young scientific and technical intelligentsia and councils of young scientists and specialists in organizing the teaching of the fundamentals of computer technology to children and young people.

The training of teachers, specialists in the methodology of teaching and organizers of computer technology and programming schools and clubs must become an important task for Komsomol organizations. It must not be expected that the higher and secondary specialized schools will provide us the necessary specialists in this field already in the next few years. For these purposes it is necessary to make extensive use of departments of social professions, schools for young scientists and specialists, universities of technical knowledge and regularly active seminars and courses. For the more effective acquaintance of children from an early age it is necessary to widely enlist young scientists and specialists in the development of computer systems and logic games and designers of video games, electronic toys and trainers which will emotionally attract a child--which would tell tales, draw, compose music and help to reveal and develop his abilities.

Specific measures have already been designated at many past reporting-and-election Komsomol meetings and conferences of scientific research institutes, design bureaus and enterprises. For example, Komsomol members of the Shipbuilding Works imeni A.A. Zhdanov came out with the initiative of launching a movement of engineering and technical personnel for training young workers in the fundamentals of working with computer equipment and training them to attend to flexible automated production processes. The initiative of the shipbuilders was supported by Komsomol members of a number of enterprises in Leningrad.

In a word, enormous tasks confront the Komsomol. The job of each Komsomol organization is to determine the specific forms of its participation in the development of computer technology and in the mastery by youth of the fundamentals of its use.

8831
CSO: 1863/126

MINPRIBOR'S ROLE IN COMPUTER ADVANCEMENT PROGRAM VIEWED

Tallinn SOVETSKAYA ESTONIYA in Russian 8 Jan 85 p 3

[Article by R. Akhmetov, correspondent]

[Abstract] The article is an interview with G. I. Kavalerov, deputy minister of instrument building, means of automation and control systems. He talks about the positions from which plants of his ministry (Minpribor) are prepared to assist in the recent party-endorsed nationwide program for the advancement of computer technology and its efficient use in the period up to the year 2000.

Kavalerov reports that Minpribor overfulfilled all of its computer-technology and control-system production assignments in 1984. The ministry's plants are meeting steadily growing demand of the Ministry of the Machine Tool Building and Tool Industry for numerical programmed control devices for machine tools and robots. N/C systems produced by Minpribor are said to meet most international standards. The introduction of a new component base for N/C devices has made it possible to reduce their weight and size by almost one-half, and to heighten their reliability and expand their capabilities, according to Kavalerov.

The development, on the basis of computers produced by Minpribor, of systems for the control of production processes (ASUTP) and systems for the automation of design work (SAPR) is called a leading direction of the ministry's work. Kavalerov notes that substantial progress has been made in the development of highly effective control complexes. In this connection, he praises the quality of two new computers: the SM-1210 and the SM-1420, which is said to be comparable to the best world models in its class. Both computers are said to be highly reliable and three to four times as productive as their predecessors. They are intended for controlling complex processes in the chemical industry and nuclear power engineering.

Kavalerov notes that more than 400 ASUTP are under development during the current five-year plan. About half of these systems are lead models with a wide range of applications, which will be produced in copies for enterprises of various types. Agreement reportedly has been reached on a national program for the development of SAPR. It calls for Minpribor to produce 8,000 automated designers' work places for various purposes in the next five-year plan.

Among other goals cited by Kavalerov for this period are doubling the production of computers; substantially increasing the supply of microcomputer-based N/C equipment for machine tools and robots; creating new enterprises which will develop software to plants' orders; and expanding the system for the technical maintenance of computers produced by Minpribor and used by enterprises of other ministries. More than 100 cities are to be included in this system.

FTD/SNAP
CSO: 1863/184

HARDWARE

COMPUTER TECHNOLOGY ADVANCES EXHIBITED

Moscow IZVESTIYA in Russian 21 Nov 84 p 2

[Article by A. Longinov: "Exhibitions; Man and Computer: Dialogue Continues"]

[Text] A multicomponent computer system distinguished by particularly high speed and making it possible to solve several programs simultaneously became a new exhibit in the "Computer Technology" pavilion of the USSR Exhibition of Economic Achievements.

This system unites 50 large units of electronic equipment operating in a 2-processor arrangement. When one of the units fails, the other is automatically switched on, continuing to run the assigned program. It is possible to make repairs or perform preventive maintenance without stopping the entire system.

It is intended for controlling large industrial systems. There are about 600,000 microcircuits in the computer system's "team."

Today computers can do thousands of quite diverse and very difficult tasks. Of the exhibits presented in the pavilion, one's attention is attracted by automated stations for producing design documentation, operating on the basis of a YeS 7055-01 and YeS 7055-02 computer, the "Granit 02-01" control console for an automated casting system, and an automatic information system for a village.

Electronic machines are continuing to learn to work more efficiently and more precisely, the exhibit relates. But it does assert that alongside every computer system, automatic system or computer there must be a human being without whom this equipment will be at a loss in the face of the first unforeseen situation.

8831
CSO: 1863/126

BASIC FEATURES OF THE SM-1600 MINICOMPUTER

Moscow VESTNIK STATISTIKI in Russian No 10, Oct 84, pp 33-35

[Article by V. Popkova, design engineer, Vilnius Calculator Factory imeni V. I. Lenin]

[Text] Since 1983, users have been receiving the SM-1600 minicomputer, which combines the capabilities and advantages of control machines (such as the SM-4) and problem-oriented machines used for processing statistical information such as the M-5000 (M-5010, M-5100) PVK perforation computer complex). The architectural and structural design of this minicomputer is based on the conceptions and integrated design element base of the SM computer.

SM-1600 structural features are: two processors--master processor (VP) and specialized processor (SP); I/O based on the "common bus" principle (a unibus interface, as opposed to the radial principle used for the M-5000). The master processor controls the computational process and I/O; the specialized processor implements the M-5100 command set, somewhat expanded (commands have been added that are essential for the concurrent operations of the two processors). This makes it possible to organize the computational process for this minicomputer in two basic ways--under the control of M-5100 DOS or one of the SM operating systems. The first way is characterized by complete software and data compatibility with M-5000 series systems and maximum efficiency when executing custom programs, and the second way allows for execution of SM-4 user programs, which must conform to SM operating system specifications, in the same way as data on its peripheral devices.

Magnetic disk (MD) is used to hold the operating system. Transfer from the control of one operating system to another on the SM-1600 is made by simply changing the operating disk and all other devices containing data that reside on magnetic disk and magnetic tape, excluding those used for storing working files or temporary data.

Basing the SM-1600 on the integrated use of master and specialized processors results in a problem-oriented SM computer that is highly efficient in solving statistical and economic planning problems.

The SM-1600 disk operating system provides complete software and data compatibility with the M-5000 series. This system resides on an SM-5408 magnetic disk. The SM-1600 disk operating system consists of M-5100 disk operating system programs; the specialized processor handles internal processing and possesses a set of M-5100 commands augmented with four additional commands: PIPS--route from "supervisor" storage (data from the "supervisor's" region may be routed to any partition); PVPS--route to "supervisor" storage (from any partition to the "supervisor's" region); PIU--check and set (for controlling the concurrent use of a region of memory by two processors); VVP--call the master processor (the primary means of calling from the specialized processor to the master processor). Calls to peripheral devices are made by means of the appropriate requests to the master processor, which, after its initial loading, waits for requests for I/O or for the execution of other functions (reader, clock, and others) from the specialized processor. Meanwhile, the specialized processor is run by the control programs and, after the execution of functions that have to do with system startup, starts accepting commands from the command input device.

Initial loading of the system is accomplished with the help of the specialized processor's hardware-implemented loader: data is read from the 0-1 sectors of the 0 track where the system resides and is brought into main memory (OP) by the control program, and the master processor is set to wait for requests.

SM-1600 hardware implementation has a number of specific features that impose certain restrictions on the implementation of concurrent operations of the two processors. A 256 K-byte main memory consists of four 64 K-byte modules, and for this reason the specialized processor may directly call only one of them at a time, i.e., it is not possible for a program residing in one of these memory modules to directly transfer control to a point located outside the module or in a similar way send data. The specialized processor is connected to the common bus in the same way as any I/O device, and all transactions between the processors take place in a pattern similar to that of peripheral device service, although the specialized processor has a separate port to main memory and may swap data, bypassing the common bus.

All operating system programs may be divided into two large groups: control and processing. The former comprise the basis of the SM-1600 disk operating system and are designed to control the operations of this minicomputer. Their functions are: prepare the operating system for operations; job entry and preparation for execution; control of program execution, as well as I/O procedures. Processing programs aid the process of compiling, debugging and storing user programs: they translate these programs from various programming languages and prepare them for machine execution, maintain the libraries, sequence or merge files, organize the exchange of data between the various storage media or devices, and conduct on-line checks of I/O devices.

The SM-1600 disk operating system makes possible multiprocessing operations with a fixed number of partitions--up to six: a background partition (FR) and give foreground partitions (R1-R5), each of which may contain only one program at a time that is executing. Each partition has its own job stream, which is not tied to and is independent from the job stream of any other partition. Resource distribution is fixed, and processor time is fixed with priority for triggering of peripheral devices and interrupts. SM-1600 time multiplexing is performed by a special module of the "supervisor" for program interrupts. Processor time is allocated to a program according to the priority of the partition in which it is being executed (R5 has the highest priority, and FR has the lowest).

In order to limit the loss of machine time in preparing to execute programs and to decrease the amount of work done by the operator, the SM-1600 disk operating system provides for batch data processing, and control commands or operators may be entered from the keyboard or read from any data device.

A;ong with a batch processing mode, the SM-1600 disk operating system offers an interactive mode, which is made possible by problem-oriented facilities that operate with the use of display terminals.

All SM-1600 operating system components (except for the "Initial Load" program), as well as user programs, are allocated to system libraries consisting of reenterable, relocatable or source modules. The "Librarian" program handles the functions of creating libraries, updating them, copying or displaying their status.

The user may write programs in assembler, COBOL and RPG in Russian and English, and PL/1 only in English, because the disk operating system includes translators that process source programs and produce relocatable modules.

The disk operating system starts its operations with the initial load procedure, which puts the minicomputer and operating system into a ready condition. During this procedure, the "Supervisor" program, which may be somewhat modified by means of the "Initial Load" program, is loaded into main memory, and then the "Job Control" program is called, which takes control.

"Supervisor" is the control program that controls the execution of problem programs from the moment of job entry to the time it produces its results. Control is possible during interrupt signals, and the program parses and processes interrupts, controls I/O, calls program phases from the reenterable module library, procedures for program and job completion, communication with the operator, clock service, creation of check points, and allocation of central processor time between programs in various partitions.

"Job Control" is also a control program and receives the batch job input stream, preparing the system to execute a job or single program. This program is called from the reenterable module library and is written into the problem program region to the partition in which the problem program is to be executed. This program selects new jobs, receives and analyzes system information, sets system parameters for a particular job, and starts the problem program's operation.

The "I/O Control" system is used for operations involving peripheral devices.

All programs executable under disk operating system control are loaded into main memory from the reenterable module library, to which they have been written by the "Editor" program, which generates programs that are ready for execution from a number of relocatable modules.

In processing economic data, tasks that include procedures for sequencing data predominate. In order to free the user from having to program these procedures, the disk operating system includes powerful facilities for sorting and merging that can operate either separately or be included in programs at the input language level.

Generation of the disk operating system for the specific user, depending on the configuration of his hardware and type of tasks to be performed, is accomplished by the "System Generation" program, and a number of special internal functions necessary for everyday operations are performed by special programs.

Thus, software for the minicomputer that comprises the SM-1600 disk operating system provides complete software and data compatibility with the M-5000 type PVK, which confirms the SM-1600 as a direct descendent of that series.

12713
CSO: 1863/154

UDC 681.32

EFFECTIVE MICROINSTRUCTION PIPELINE ORGANIZATION IN A HIGH-CAPACITY MICROPROCESSOR

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 3, July-Sept 84 (manuscript received after revision 25 Jan 84) pp 38-41

[Article by V. V. Novoselov, candidate of technical sciences, under the rubric: "Automation and Mechanization in Control Systems"]

[Text] The efficiency of a microprogrammable processor (MP) with a fixed structure and a fixed means of processing microinstructions is defined in terms of execution times for microprograms making up a given microprogram control algorithm and characterizes the degree of correspondence between MP structure/microprogram processing means and algorithm characteristics. Therefore, different algorithms and even segments of a single algorithm require different MP structural and processing solutions. Thus, an increase in the extent of pipeline handling permits a reduction in timing cycle duration, but also leads to an increase in the number of cycles required for the execution of conditional branches resulting from a prior operation. In other words, the MP structure with the maximum degree of pipeline handling for microinstruction processing is more effective for linear algorithm sections while segments with conditional branches are handled more effectively by an MP structure featuring sequential microinstruction processing.

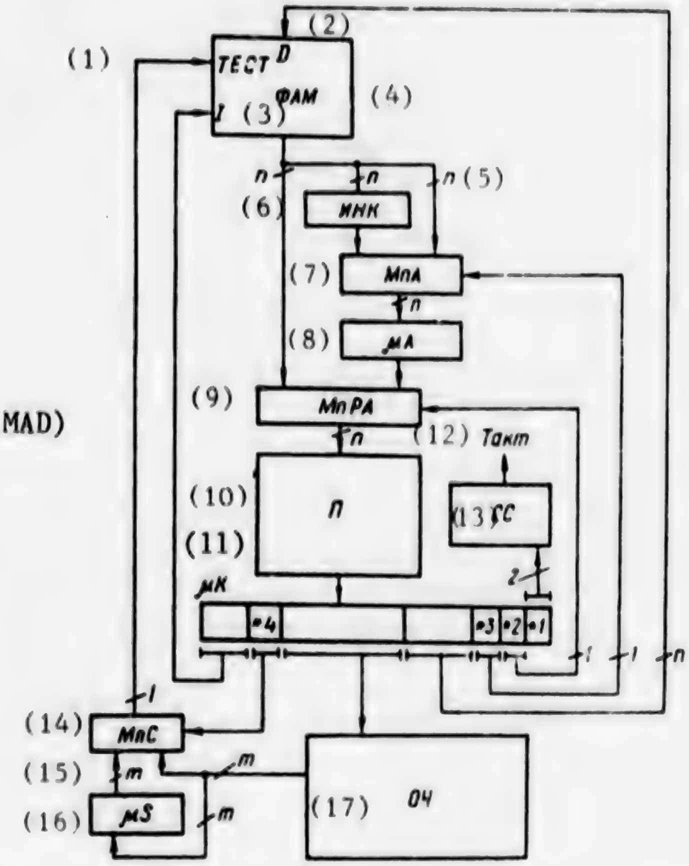
An MP structure/instruction processing design solution called the adaptive microinstruction pipeline has been developed to assure maximum efficiency in the execution of both sequential and arbitrary algorithm segments.

The adaptive microinstruction pipeline operates on sequential sections with a cycle duration of $t1 = \max(t_{OS}, t_M, t_{MAD})$.

In executing conditional branches as a result of a prior operation the cycle duration is $t2 = t_{OS} + t_M + t_{MAD}$ (here the branch to the correct address occurs entirely within one cycle).

Finally, in executing conditional branches according to previously formed conditions the cycle duration is $t3 = \max(t_{OS}, t_M + t_{MAD})$, where t_{OS} is the duration of the operation in the MP's operational section (OS), t_{MAD} is the duration of the operation in the microinstruction address driver (MAD), and t_M is the microprogram memory access time.

The structure of the adaptive microinstruction pipeline is shown in Fig. 1. A microprocessor large-scale integrated circuit (LSIC) without an address register at the output (e.g. the K1804VU1, K1804VU2 or K1804VU4) can be used as the MAD. All registers shown in the figure operate on the leading edge of the clock pulse and can be implemented with the K1804IR1 circuits incorporated in the K1804 microprocessor LSIC unit.



1 - TEST (condition input)
2 - Address input
3 - MAD instruction input
4 - Microprocessor address driver (MAD)
5 - n (address word length)
6 - Incrementor (INC)
7 - Address register multiplexer
8 - Address register (AR)
9 - Resultant address multiplexer
10 - Memory
11 - Microinstruction register
12 - Clock cycle
13 - Clock circuit
14 - Condition multiplexer
15 - m (flag bus word length)
16 - Condition register (CR)
17 - Operational section (OS)

Fig. 1. Adaptive microinstruction pipeline block diagram

Adaptive microinstruction pipeline operation during each cycle is defined by microinstruction fields marked with the symbol "-". The contents of these fields are specified in accordance with one of the following types of microinstructions:

1. A microinstruction whose address portion directs an unconditional branch.

In a flow chart (Fig. 2), this type of microinstruction corresponds to a block of operators O_1, O_2, O_6 which is in turn followed by an operator block.

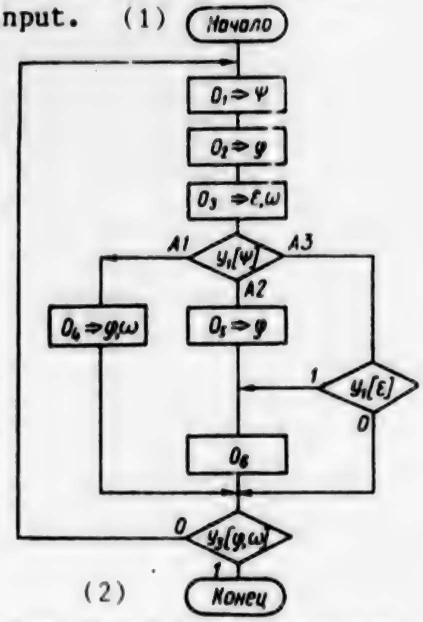
The contents of all microinstruction fields are shown in Table 1 which illustrates in symbolic form a microprogram resulting from a given control algorithm (Fig. 2) implemented on an adaptive microinstruction pipeline.

Table 1 - Algorithm microprogram

	Microinstruction					
Address	Operational Section	Address Section	*1	*2	*3	*4
i	$O_1 \Rightarrow \psi$	Jump to i+2	t1	AR	MAD	X
i+1	$O_2 \Rightarrow \psi$	X	t1	AR	X	X
i+2	$O_3 \Rightarrow \varepsilon, \omega$	$y_1[\psi]$ $<i+3, i+4, i+6>$	t3	MAD	INC	CR
i+3	$O_4 \Rightarrow \varphi, \omega$	$y_3[\varphi, \omega]$ $<i+8, i>$	t2	MAD	INC	OS
i+4	$O_5 \Rightarrow \varphi$	X	t1	AR	X	X
i+5	O_6	$y_3[\varphi, \omega]$ $<i+8, i>$	t3	MAD	INC	CR
i+6	NOP	$y_2[\varepsilon]$ $<i+5, i+7>$	t3	MAD	X	CR
i+7	NOP	$y_3[\varphi, \omega]$ $<i+8, i>$	t3	MAD	INC	CR
i+8	Halt	X	t1	X	X	X

Note: X = value not important.

Thus, in the execution of this first type of microinstruction, during the current cycle an operation is executed in the OS, designating the operational field of the given microinstruction. The next microinstruction will be fetched from memory and placed on the microinstruction register input and the address of the microinstruction assumed as the next to be executed will be placed on the address register input.


(1)
Начало
$O_1 \Rightarrow \psi$
$O_2 \Rightarrow \varphi$
$O_3 \Rightarrow \varepsilon, \omega$
A1
$y_1[\psi]$
A3
A2
$O_4 \Rightarrow \varphi, \omega$
$O_5 \Rightarrow \varphi$
1
$y_1[\varepsilon]$
0
O_6
0
$y_3[\varphi, \omega]$
1
(2)
Конец

1 - Beginning
2 - End
O - Operator block
Y - Condition block

$O_1 \Rightarrow \psi$ - formation of condition ψ by operator block O_1; $Y_1(\psi)$ - testing of condition ψ by condition block Y_1.

Fig. 2. Example of a flow chart for a microprogram control algorithm

2. A microinstruction whose address portion directs a conditional branch according to the value of the flag set by the previous operation in the OS.

In a flow chart, this type of microinstruction corresponds to conditional and previous operator blocks in which the operator block forms the value of a flag tested by the conditional block. In Fig. 2 this microinstruction corresponds to blocks (O_4 and Y_3).

As a result, during the current cycle an operation is executed in the OS which designates the operational field of a given microinstruction, the branching condition is developed and the address of the next instruction is calculated and fetched to the microinstruction register input. The address of the microinstruction assumed to be the next is set on the address register input.

3. A microinstruction whose address portion specifies a conditional branch according to the value of a flag set in the OS by an operator block, which is not the previous condition block, defining a given conditional branch.

On a flow chart, this type of microinstruction corresponds to conditional and previous operator blocks in which the operator block does not set the value of the flag tested by the conditional block. In Fig. 2, these are the pairs (O_3, Y_1), (O_6, Y_3), (NOP_1, Y_2) and (NOP, Y_3). The symbol NOP stands for no operation, i.e. the fact that no operation takes place in the OS during a given cycle.

As a result, during the current cycle an operation is executed in the OS which specifies the operator field of a given microinstruction, while at the same time, the address of the next microinstruction is calculated and placed on the microinstruction register input. The address of the microinstruction assumed to be the next to be executed is placed on the address register input.

Figure 3 illustrates the dynamics of adaptive microinstruction pipeline operation during the execution of the control algorithm given.

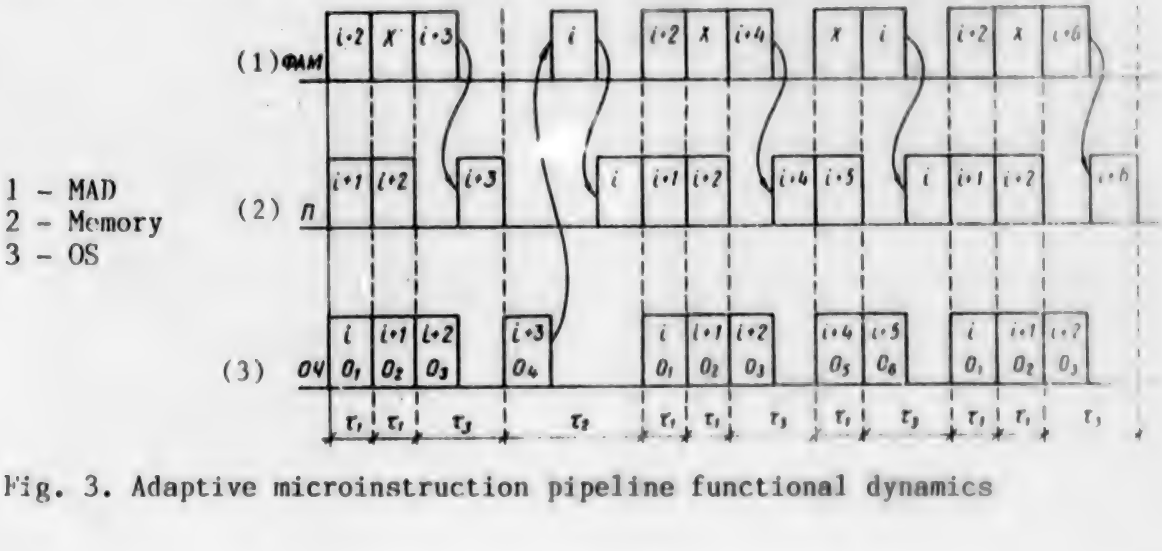


Fig. 3. Adaptive microinstruction pipeline functional dynamics

Thus, by organizing a microinstruction pipeline in the manner described it is possible to work in either the pipeline or sequential modes, according to the characteristics of the algorithm section executed, as a result of MP structure adaptation during the course of microprogram execution.

Table 2 - Comparison of MP structural efficiency

Algorithm type, Operand length	Components	MP structure	Rel. T	Rel. H
Floating point multi-	K1804VU1,	S_1	1.67	1.00
plication. 24-place	K1804VU3,	S_4	1.23	1.16
mantissa. Cycle pointer	K1804VS1,			
implemented in OS.	K1804VR1,	S_7	1.12	1.20
	K1804IR1,	Sap	1.00	1.43
	K565RU2			
Floating point multi-	K1804VU4,	S_1	1.67	1.00
plication. 24-place	K1804VS2,	S_4	1.09	1.18
mantissa. Cycle pointer	K1804VR1,			
implemented in MAD.	K1804IR1,	S_7	1.31	1.22
	K565RU2	Sap	1.00	1.46
Sum of squares of	K1804VU4,	S_1	1.80	1.00
differences calculation.	K1804VS2,			
	K1804IR1,	S_4	1.16	1.19
$C = \sum_{K=1}^{M} (A_K - B_K)^2$	K565RU2	S_7	1.22	1.24
Operand length A_K, B_K - 1 byte. M = 16		Sap	1.00	1.50

Table 2 shows the characteristics of various implementations of the adaptive microinstruction pipeline on various K1804-series microprogrammable microprocessor LSICs. The relative microprogram execution time (Rel. T) for the adaptive pipeline (Sap) is assumed to be 1.00 and is compared to the execution times for the same algorithms on MPs with different structures (S_1, S_4, S_7).* The relative hardware ratio (Rel. H) reflects the number of small- and medium-scale integrated circuits required to produce each MP

* Novoselov, V. V., Shumilov, L. A., "Vybor Struktury Mikroprotsessora na Komplekte Mikroprogrammiruyemykh BIS" [Selection of Microprocessor Structures Based on Microprogrammable LSIC Units], USiM, 1983, No 3, pp 21-24

structure. Using the three algorithms indicated, the adaptive microinstruction pipeline achieved a 9-80 percent increase in efficiency while increasing the hardware requirement by 16-50 percent. The number of microprogram memory LSICs (i.e. microinstruction word length) increases by an overall 2-5 percent, regardless of the algorithm type.

12746
CSO: 1863/66

UDC 621.317.757

SPECIAL-PURPOSE PROCESSOR FOR DIGITAL FREQUENCY ANALYSIS OF SIGNALS

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 1 Feb 84) pp 26-28

YEFIMOV, V. M., ZOLOTUKHINA, M. A., KUZNETSOV, V. V. and YAKUSHEV, V. S., Novosibirsk

[Abstract] The special-purpose processor for digital frequency analysis of signals (S. V. Bredikhin in AVTOMETRIYA No 4, Jul-Aug 84 pp 20-25) operates in real time. It can simultaneously filter signals in L = nm frequency bands during (1/m)-octave analysis in n octaves at a rate as high as $F_{max} = \alpha f_{max} nm$ (f_{max}- upper limit of frequency band, $\alpha \geq 2$ matching factor between discretization frequency and cutoff frequency of theanalog low-frequency prefilter). As the discretization frequency may be lowered for lower frequencies in the signal spectrum, the processor structure can be reduced to a linear array of digital band-pass filters, a linear array of digital low-pass filters, and a linear array of averagers containing square-law detectors. Digital filters are synthesized to specifications by connecting two-pole resonator circuits in series, the standard technology being the same as for analog filters. Such a circuit must, for producing a readout, perform generally six multiplications and two summations. The basic components of such a circuit are arithmetic devices, namely three multipliers and two summators operating either simultaneously or sequentially. The processor can perform (1/1)-octave and (1/3)-octave analysis in the 0-11.2 kHz frequency band followed by (1/12)-octave analysis, analysis in low-frequency and infralow-frequency ranges is facilitated by lowering the cutoff frequency of the analog prefilter and correspondingly the frequency of the clock generator. Figures 2; references: 4 Russian.
[74-2415]

UDC 681.518.3

BASIC CONFIGURATIONS OF 'MICRO-CAMAC-LAB' SYSTEMS

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 11 Mar 84) pp 15-20

GUSEV, O. Z., ZOLOTUKHIN, Yu. N., PROKHOZHEV, O. V. and YAN, A. P., Novosibirsk

[Abstract] The three basic configurations of micro-CAMAC hardware and software developed in the Siberian Department of the USSR Academy of Sciences are a local one for laboratory test systems, a terminal one for data gathering and experiment control systems, and an autonomous one for data processing systems which use rigid programs for implementation of laboratory or technological procedures. The essential equipment common to all three configurations is an "Elektronika-60" microcomputer with direct-access memory, interfacing, and high-speed printer, an alphanumeric display, a CAMAC crate with power supply, a crate monitor acting as program-controlled interface between crate bus and microcomputer, a graphic raster display, a color raster display, a graph plotter, and a bus indicator. In the autonomous configuration is included a programmer of microcircuits in the direct-access memory, specifically series K155RYe3, KR556RT4, and KR556RT5 microcircuits. All documentation pertaining to these three "Micro-CAMAC-Lab" configurations has been transmitted from the participating research institutes to the pilot production plant of the USSR Academy of Sciences. Figures 4; references: 7 Russian.
[74-2415]

UDC 681.324

HIGH-PRODUCTIVITY PERIPHERAL VECTOR PROCESSOR A-12

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 25 Feb 84) pp 29-35

BRODSKIY, I. I., KOZLACHKOV, KORSHEVER, I. I., NESTERIKHIN, Yu. Ye., PAVLOV, S. A. and REMEL', I. G., Novosibirsk

[Abstract] A peripheral vector processor A-12 has been developed at the Institute of Automatic Control and Telemetry in the Siberian Department of the USSR Academy of Sciences, with software equivalent to that of the Floating Point Systems Co.'s model AP-120B. It is built entirely with Soviet-made microelectronic components. Its two arithmetic devices, namely an adder and a multiplier with a floating point each, operate at a rate of 10^6 floating-point operations per second. The clock frequency is 6 MHz. The processor incorporates supercomputer technology, economically matched to the base computer, using emitter-coupled logic and transistor logic with Schottky-barrier diodes. The basic architecture is that of synchronous-conveyor data

processing, with three conveyor steps in the multiplier and two conveyor steps in the adder. It has a direct-access memory, a tabulation memory, and a cache in a multilevel configuration with tuning input and output multiplexors. An array of interconnected data buses enhances the programming flexibility. Programming for solution of application problems is done with an Assembler, with a mathematical library, or with a compiler from high-level language such as FORTRAN. The practically attainable productivity of this processor depends largely on the software and can be optimized to 60-80% of the "peak" productivity when operating with the Assembler, but only to 20-40% when operating with the compiler from FORTRAN and to 50-20% when operating with the mathematical files. There has also been provided a software for program tracking and debugging. Like the "Elektronika" family of small computers, this processor can be controlled by a real-time operating system or by a RAF [expansion not given] operating system. It performs calculations for typical problems six times faster than a BESM-6 high-speed computer. Figures 3; references 13: 4 Russian, 9 Western.
[74-2415]

UDC 681.327.22:621.397.6.037.733.2

THREE-PORT VIDEO FRAME BUFFER

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 13 Jan 84) pp 76-80

KOVALEV, A. M., KUROCHKIN, V. V. and TARNOPOL'SKIY, Yu. V., Novosibirsk

[Abstract] A video frame buffer with three ports is synthesized for processing of aero-space data so as to match the sweep rate to the variable rate of data input from the video processor array. Port A connects the buffer to the image source or the video processor. Port B displays the buffer contents on the television monitor screen. Port C connects the buffer to the computer. The number of buffer cells is a multiple of both 16 and 24, matching the X;Y= 4:3 frame format, all cells mounted on 10 multilayer printed-circuit boards. The buffer has an input bus and an output data bus as well as an address bus. A control and cynchronization module generates not only "read/record" but also RAS and CAS time signals. The video port B contains a digital 20-to-1 commutator, a γ-correction memory which stores 8-digit codes for each color component and converts them to 11-digit codes so as to compensate the video receiver nonlinearity, a port synchronizer, and a video format control. Port C is a direct-access interface. Through port A, which is of third-level priority, dynamic scenes are transmitted from the video processor to the television monitor. Every third buffer cycle is assigned to operation with the video port, and thus two cycles are available for loading the buffer with data from the video processor. This can be done at a rate as high as 6 Mbit/s. Figures 3; references 22: 9 Russian, 13 Western (1 in Russian translation).
[74-2415]

UDC 681.327.23

COLOR GRAPHICS DISPLAY

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 28 Dec 83) pp 81-85

OSTAPENKO, A. M. and SHEMETOV, S. A., Novosibirsk

[Abstract] A color graphics display has been developed as an aid in designing the topology of multilayer printed circuits and large-scale integrated circuits. Its base is a KR580IK80 microprocessor controlling the entire operation and capable of operating with the computer in the dialog mode with either symbolic or graphic data, also simultaneously with both kinds of data from two independent terminals. For the symbolic mode it has available 80 symbols and 16 lines. For the graphic mode it has a 256 kbyte - 4 Mbyte memory. The screen size is 320x256 points, the memory has a 128-2048 vertical field and a 512-2048 horizontal field. The display components in addition to the microprocessor and a graphic memory with control and display shaper are a cursor shaper, a mask chart, a color chart, and a symbol module. The hardware functions have been minimized by maximizing the software, the symbolic part of the display being programmed to peform all functions of the "Videoton-340" display. The entire display program is subdivided into processing external instructions and executing the display algorithms. The display can operate on-line with a master computer as well as autonomously. When operating in the graphic mode, the SHIFT keyboard not only performs its conventional functions but also "shifts" the screen in the memory. The display is operating at the Institute of Automatic Control and Electrometry, Siberian Department of the USSR Academy of Sciences. Figures 2; references 2: 1 Russian, 1 Western.
[74-2415]

UDC 629.7.058.74:681.3.06

SCANNING GRAPHICS DISPLAY FOR TWO-COORDINATE DESIGN SYSTEMS

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 23 Jan 84) pp 85-89

KOVALEV, A. M. and TAKNYKIN, E. A., Novosibirsk

[Abstract] A scanning graphics display has been developed for operation with an "Elektronika'60" microcomputer as processor and an SM-4, "Elektronika-100/25" or "Elektronika-79" master computer, for the design of printed circuits. Its key component is a memory which consists of six 512x512 bit banks on two or three cards, 16-digit words being used for data exchange with the address register. Screen scaling is possible, from a scale register according to a special program. An X-Y locator whose position on the screen is controlled by two registers facilitates operation in the graphic dialog mode. When the memory is split into three cards, with two banks on each and

each card corresponding to a bit image, then one would assign to them colors red, green, blue respectively and make the locator whie (red+green+blue). A different color assignment would be appropriate in the case of two cards with three banks on each. Another component of the display is a control register which turns images on or off, turns the locator on or off, permits or forbids intracard between banks interruption, and determines in which way the memory is to be split. The necessary software consists of internal modules, allowing application of users' programs entered either into the microcomputer or into the master computer. There are already more than 40 routines, including those which switch the display into the textual mode and into the graphical mode. Figures 2; references: 3 Russian.
[74-2415]

UDC 681.327.22

COLOR TELEVISION CAMAC-DISPLAY

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 31 Jan 84) pp 93-96

YAKUSHEV, V. S., Novosibirsk

[Abstract] A CAMAC display module for driving color television monitors is described which matches a frame format of 384x256 points. It ensures independent displaying of the memory contents on the screen and recording of new data in the memory. It also ensures independent access to the various layers of the video frame buffer. It is technologically feasible in the form of a module of unary width and provides for refining the gradation of image brightness. Each of its three 96 kbit two-port direct-access memories on a common multilayer printed-circuit board stores brightness and color information bits about all points of a frame, with multiplexing of the address bus, the video ports connecting them to the monitor and the program channel ports connecting them to the computer crate bus. The module has three digital outputs to color control channels, one sweep synchronization output, and one analog output for semitone signals. A timer controls the operation cycle and the data display, in synchronism with the television sweep. The software contains nine instructions necessary for operation. Three bits are used for coding the eight basic colors (000 black, 001 dark blue, 010 green, 011 light blue, 100 red, 101 purple, 110 yellow, 111 white), with digital-to-analog conversion of this "subcode" through a GRAY IN - GRAY OUT channel for black-white gradation signals. This color television drive is widely used in "MicroCAMAC-Lab" systems, particularly in the "Yenisey" digital frequency analyzer. Figures 3; references: 5 Russian.
[74-2415]

UDC 681.335

IMPROVING ACCURACY OF CALCULATORS WITH EXPONENTIAL SWEEP

Leningrad IZVESTIYA VYSSHIKH UCHEBNYKH ZAVEDENIY: PRIBOROSTROYENIYE in Russian No 10, Oct 84 (manuscript received 27 Jan 84) pp 37-42

BERESNEV, V. K. and ROGACHEVSKIY, B. M., Novosibirsk Institute of Electrical Engineering

[Abstract] Calculators operating with instantaneous values of exponential sweep voltages are accurate within 0.15-0.3% and need a time of 1-3 ms for multiplication, division, or raising to a power. Major sources of inaccuracy and sluggishness are delay of the comparator response and instability of time constants of the exponential-voltage generators. A way to reduce the inaccuracy is to compensate the time delay, by converting the comparator input and output voltages simultaneously and similarly into time intervals, then subtracting the converter signals and letting the difference signal control the fast recharge of the analog storage. Meanwhile, the effect of instability of time constants on calculation of powers can be reduced appreciably by comparing the input voltage V_1 with the exponential voltage V_0e^{-t/T_e} and converting it into a corresponding time interval $\tau_1 = T_e\log_e(V_0/V_1)$, then changing the duration of comparator output pulses to $\tau_2 = mT_e\log_e(V_0/V_1)$ and changing the output voltage till it becomes equal to $V_0(V_1/V_0)^m$, where m is the exponent of the sought power function. Now a 15-20% variation of a time constant will distort the result of calculation by not more than 0.1% and the error of squaring will be of the order of 0.05% over the entire 0.4-4.0 V range of input voltage. Article was recommended by Department of Information and Measurement Engineering. Figures 4; references: 4 Russian.
[90-2415]

UDC 519.713

PERFORMANCE ANALYSIS OF MULTIMACHINE COMPUTER SYSTEMS

Riga AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA in Russian No 5, Sep-Oct 84 (manuscript received 15 Mar 83, after revision or completion 19 Sep 83) pp 46-52

LEVIN, V. I.

[Abstract] A method is proposed for deterministic performance analysis of multimachine computer systems. It is based on the mathematical apparatus originally developed by the author for analyzing the dynamics or logic devices and systems (V. I. Levin, Izd-vo Energiya, Moscow 1980). It is demonstrated on the simple model of a few operating computers, but it is applicable to any number of computers solving any number of problems. The mathematical apparatus includes an infinite-valued logic and a dichotomic procedure, the latter reducing to a sequential breakdown of the resolvent determinant into smaller

ones by disjunction of conjunctions with a subsequent ordering procedure. The fundamental relation characterizing the system performance is defined so as to reduce evaluation of the system performance to evaluation of a time vector: instants of time at which the computer system has solved each problem. The algorithm of this calculation is constructed first for a homogeneous computer system and then for a nonhomogeneous one, in accordance with a corresponding theorem regarding the appropriate recurrence relation for each case. The volume of calculations necessary for evaluating this time vector (matrix) is also calculated. The algorithm of this performance analysis has, moreover, been programmed in FORTRAN for a YeS-1022 computer. Typical results are shown for a system of m= 7 computers solving n= 10 problems and a system of m= 9 computers solving n= 10 problems. References 3: 2 Russian, 1 Western (in Russian translation).
[104-2415]

UDC 681.324:519.713

EFFICIENCY AND PERFORMANCE OF MULTIMACHINE AND MULTIPROCESSOR COMPUTER SYSTEMS

Riga AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA in Russian No 5, Sep-Oct 84 (manuscript received 17 Nov 83) pp 53-59

KONONOV, N. V. and PESHES, L. Ya.

[Abstract] An analytical method of evaluating the time characteristics of multimachine and multiprocessor computer systems is described, this method yielding more precise results than existing empirical ones or simple summation. It is based on the model of a system with a finite number of computers, each of them subject to failure and restoration during the problem solving process, but at least one of them operational at any one time. The operation of this system is regarded as a semi-Markovian process with continuous time and discrete states. The mean problem solution time is estimated for two variants of system operation. In the first case none of the computers which have failed is restored to operation in time for solving the problem, except the last one if it was the only one left. In the second case all the computers which have failed are restored in time for participation in the problem solving process, if this can be done before completion of that process. The system of equations for the mean time is solved by statistical simulation. The efficiency $0 \leq \beta \leq 1$ is then calculated as function of the number of computers in the system, assuming a homogeneous system and disregarding the reliability factor. The results are useful for optimizing a multimachine computer system, upon determination of the number of computers beyond which there will be no improvement of performance and by using a benchmark mix of problems so as not to bias the system capability. Figures 3; references: 1 Russian.
[104-2415]

UDC 681.3:621.396.535.8

ORGANIZATION OF CHANNEL SWITCHING IN PARALLEL COMPUTER SYSTEMS WITH OPTICAL ARRAY COMMUTATOR

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, Sep-Oct 84 (manuscript received 8 Jan 82) pp 16-23 and 39

GOLOMIDOV, YURIY VITAL'YEVICH, engineer, Leningrad, LI, SI KEN, doctor of technical sciences, docent, Leningrad Institute of Water Transportation, POPOV, STANISLAV ALEKSANDROVICH, doctor of technical sciences, department head, Leningrad Institute of Water Transportation, and SMOLOV, VLADIMIR BORISOVICH, doctor of technical sciences, department head, Leningrad Institute of Electrical Engineering

[Abstract] An optical array commutator is proposed for channel switching in parallel multiprocessor computer systems, allowing the rearrangement (channel to central processor, channel to main memory, channel to system monitor) with maximum processing efficiency and structural economy. Various existing electronic and optoelectronic switching devices are comparatively evaluated in terms of technological and performance characteristics, their classification on the basis of these indicators revealing the advantages of an optical array commutator. It is fully accessible and nonblocking, it can be combined with a control device, one input being connectible to several outputs through a sequence of points to each. The commutator performs not only the operation "switch" but also the operation "pairwise exclude", namely excludes intersections while operating in the mode of paginal data processing. Its capabilities, namely speed and cyclicity in addition to decoupling reliability, make it particularly suitable for organizing parallel operation in a conventional computer system. Figures 6; references 17: 10 Russian, 7 Western (1 in Russian translation).
[106-2415]

UDC 681.3.06

ANALOG-DIGITAL GENERAL-PURPOSE COMPUTER SYSTEMS BASED ON AVK-3 SMALL ANALOG COMPUTERS

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, Sep-Oct 84 (manuscript received 2 Jul 82) pp 40-43

BELYAKOV, VITALIY GEORGIYEVICH, candidate of technical sciences, department head, and VITENBERG, ISAAK MOISEYEVICH, department head, Scientific Research Institute of Computing Machines, Moscow, SVYATNYY, VLADIMIR ANDREYEVICH, candidate of technical sciences, docent, and FEL'DMAN, LEV PETROVICH, doctor of technical sciences, department head, Donetsk Polytechnic Institute

[Abstract] Programs and structures have been developed jointly by the Scientific Research Institute of Computing Machines and the Donetsk Polytechnic Institute for ATsVS-31/32/33 analog-digital computer systems consisting

of an SM-2 small digital control computer and up to 16 AVK-31 small analog computers. Each of the three systems includes a computer-object interface for transmitting eight analog quantities through analog-to-digital converters and four discrete quantities through digital-to-analog converters to any processor, input of two logic variables to an AVK-31 computer and output of two logic variables as well as two interrupt signals from an AVK-31 computer. The software for all three ATsVS variants, ATsVS-31 being used primarily for scientific research, is based on six principles: 1) decomposition of the system into functional subsystems, with analysis of both analog and digital parts as well as the communication link during all stages of data preparation and problem solution; 2) systematic organization of all subsystems' operation; 3) maximum utilization of standard programming meansfor small computers; 4) commonality and succession of programming means for all three ATsVS structures; 5) generability of software contents; 6) selection of software development and debugging sequence in subsystems so as to ensure earliest readiness of ATsVS for operation. Use of small computers accelerates generation of software, while their modular structure facilitates layout and experimental testing of ATsVS structures. Small computers allow using the digital part of software as real elements of a study object in a multiproblem situation such as automation of test programs or scientific experiments. Coupling small computers to YeS computers facilitates the design of ATsVS with hierarchical structure. Further developments underway include installation of ATsVS in enterprises, improvement of the software for these three one-level ATsVS variants, software for the new ATsVS-34 variant, and software for the ATsVS-35 two-level variant. Figures 3; tables 2; references: 4 Russian.
[106-2415]

UDC 681.327.17

DEVELOPMENT OF MICROELECTRONIC ELEMENTAL BASE FOR SIMULATING HYBRID COMPUTERS

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, Sep-Oct 84
(manuscript received 23 Jun 82, after completion 8 Apr 83) pp 82-84

PANCHISHIN, VALENTIN IGNAT'YEVICH, candidate of technical sciences, senior scientific associate, Institute of Simulation Problems in Power Engineering, UkSSR Academy of Sciences, Kiev, and BYKADOROVA, GALINA VLADIMIROVNA, candidate of technical sciences, senior scientific associate, Vornonezh State University

[Abstract] A data converter has been developed especially for network analyzers with analog processors. This "thermotron" is a functional element with a continuous spectrum of resistance variation, particularly useful for simulation of power networks. It consists of a transistor acting as power amplifier and a thermistor separated by a layer of material which is a thermal conductor but an electrical insulator. The transistor is a linear or nonlinear heat source, depending on whether it has been connected into a common-base or common-emitter circuit. Two versions of a thermotron have been built and tested. The compound thermotron consists of a planar mesa silicon transistor

and an oxide thermistor, with a very wide range of output resistance variation (1:1000) but also a wide variance of the nominal output resistance (1.1 Mohm). The laminate thermotron consists of a planar mesa silicon transistor and a silicon thermistor, with a much narrower range of output resistance variation (1:2.5) but also a negligible variance of the nominal output resistance (25 ohms). The impossibility of attaining a sufficiently small resistance close to zero and a sufficiently large resistance approaching infinity is overcome by additional control based on the field effect in semiconductors. Both thermotrons can be used in symmetric or asymmetric electrical circuits. Figures 4; tables 1; references: 9 Russian.
[106-2415]

UDC 681.320

DEVICE FOR GRAPHICAL COMPUTATIONS

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, Sep-Oct 84
(manuscript received 28 Jun 82) pp 85-87

GORELOV, ANATOLIY BORISOVICH, candidate of technical sciences, laboratory chief, TKALENKO, NIKOLAY ANDREYEVICH, candidate of technical sciences, laboratory chief, and PIVOVAROV, NIKOLAY VLADIMIROVICH, senior scientific associate, Scientific Research Institute of Automatic Planning and Management Systems for Construction Industry, Kiev

[Abstract] A device has been developed for graphical computations which facilitate numerical evaluation of variables in management of multioperational projects. The model of such a project is constructed in the form of a graph on time grid with numerical estimates of demand for production resources. The device consists of a microcalculator, a control module, a hairline slider with position recording indicator, a row number recording indicator, two column number recording indicators, and a power supply. The device can compute the distribution of production resources, the volume of project operations, the intensity of demand for production resources, and the state of production resources. The amount of calculations can be reduced appreciably by dealing only with changes in production resources as a result of graph corrections made on the basis of management data processing. The device is useful for on-site management where installation of a computer for dialog mode of operation is too expensive and access to remote computer centers is either not feasible or not reliable. Figures 1; references: 2 Russian.
[106-2415]

SOFTWARE

UDC 681.326.53

PRINCIPLES OF ORGANIZING A REAL-TIME CONTROL SYSTEM

Kiev DOKLADY AKADEMII NAUK UKRAINSKOY SSR, SERIYA A: FIZIKO-MATEMATICHESKIYE I TEKHNICHESKIYE NAUKI in Russian No 11, Nov 84

[Article by A. F. Verlan' and V. V. Galkin, submitted to the UkSSR Academy of Sciences by V. I. Skurikhin]

[Text] When designing and using software for real-time control systems, special difficulties are caused by the process of searching for and eliminating program errors. It is known that the cost of software in comparison with the cost of computer hardware is increasing constantly and by 1985 will constitute 90 percent of the total cost of computer systems, of which 50-60 percent will be spent on debugging [1]. Even after completing the development it is necessary to expend significant resources on support. According to available data, at the General Motors Corporation 75 percent of the effort related to program development goes toward maintenance under operating conditions [2].

An important feature of real-time programming is multitasking during joint use of computer system resources (processors, data, programs, peripherals) under the control of a strict time schedule and random flow of demands for service.

This presents rigid requirements for the debugging and monitoring devices to minimize the use of the resources of the real-time system, and especially during the process of conducting complex program adjustments and tests. Otherwise, the conditions for verifying programs will be very far from actual operating conditions because of the destruction of time interrelationships and the use of computer resources by indirect allocation. Such a situation does not allow verifiable conclusions to be drawn as to the quality of the program being tested, which subsequently may lead to irreversible consequences during use of the real-time system.

Existing service programs and hardware are basically oriented toward off-line debugging of individual programs [3-5] without providing the possibility of clarifying reasons for the occurence of an error and its effects on the complex debugging and tests of the programs. The development of programming for automating program tests [6] clashes with the necessity for interfering in the operation of a test program by using "traps" at control points to interrupt the actual computing process to complete service operations, which in the final analysis leads to lost time and introduces distortion into the real time scale.

The random moments of information arrival times and adjustments to include specified subprograms, the change in intensity of the stream of demands the random composition of accumulated information and solution time for each problem, the random nature of connection and the conditions of program interaction [7] lead to the fact that, in case of an outage or failure in program operation, the personnel faces the problem, difficult to solve, of diagnosing the cause and location of the difficulty. If one takes into account that the greatest errors are certain to be in time estimates for implementing different program groups and for distributing computer output [7] (in particular in a multiple machine system), then the necessity to create new means for ensuring the possibility of monitoring program functioning during the complex debugging, tests and use of a real-time system becomes obvious.

A basic unit of control in modern operating systems is the problem, which is an amalgamation of programs and data requiring resources for its execution. For every real-time problem parameters are given, such as a name, priorities, start-up time, completion interval, etc., which are stored in the problem control unit. The problem supervisor analyzes the sequence list and transfers control of the processor to the problem with the highest priority (Figure 1).

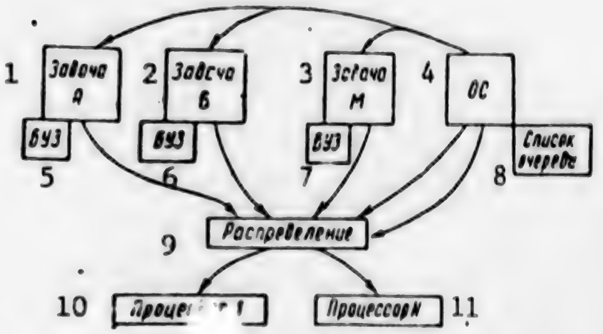


Figure 1. Logic Structure of Problem Distribution (OS - Operating System)

Key:

1. Problem A
2. Problem B
3. Problem M
4. Operating system
5. Problem control unit
6. Problem control unit
7. Problem control unit
8. Sequence list
9. Distribution
10. Processor 1
11. Processor N

To evaluate the time distribution and other resources of the computer system, during the process of comprehensive debugging and program testing it is necessary to obtain information on the time, problem inclusion sequence and their basic parameters.

For solving the indicated problems a new principle for organizing the process of monitoring program functions seems promising, consisting of the comprehensive use of hardware independent of the computer complex control system (Figure 2). Such an approach--the use of hardware--guarantees the accumulation of diagnostic information without interfering in the operation of the programs under study, and the functions of processing the derived data and implementing different monitoring methods are entrusted to its own programming.

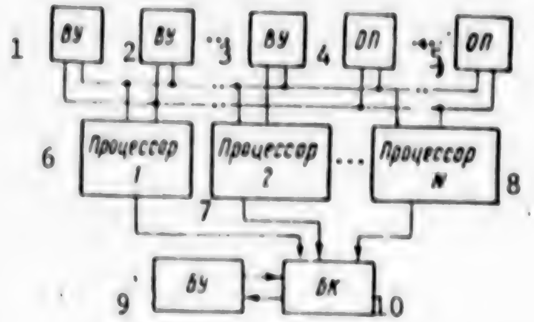
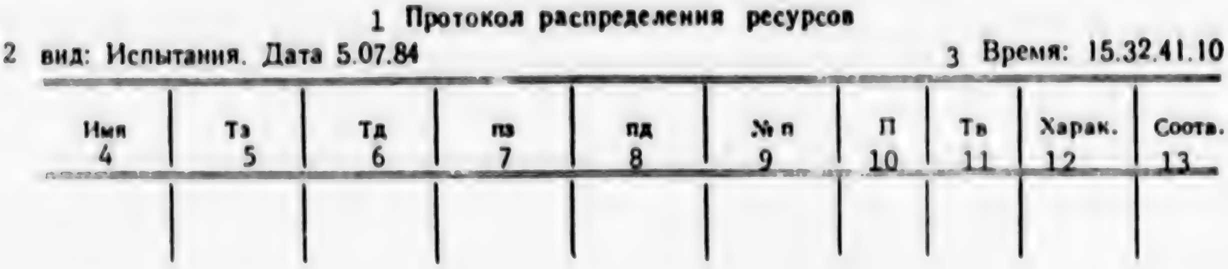


Figure 2. Structural System of Switching Device

Key:
1. Input control
2. Input control
3. Input control
4. Main memory
5. Main memory
6. Processor 1
7. Processor 2
8. Processor N
9. Control unit
10. Monitoring unit

The suggested approach ensures the possibility of obtaining the necessary data with minimal expenditure of the resources of the real-time system. For this, in the programming process a special command is added at the beginning and end of the problem (or other position in need of monitoring). Such a command guarantees the transmission of the lay-out of the main parameters of the problem (such as a minimum code name) to the monitoring unit. The monitoring unit, having determined the arrival time of the command and the number of the computer processor, ensures the temporary storage of the information obtained. Then the control unit, constructed on the basis of a microcomputer, using its own program, reads out the accumulatec monitoring unit information, determines the remaining given parameters of the problem, calculates the period and time for completion and prints an operating record for the realtime system (Table) indicating impedance of the operating schedule due to inopportune inclusion of problems to execute.



1 **Протокол распределения ресурсов**

2 вид: Испытания. Дата 5.07.84 3 Время: 15.32.41.10

Имя 4	Тз 5	Тд 6	пз 7	пд 8	№ п 9	П 10	Тв 11	Харак. 12	Соотв. 13

1. Resource Distribution Record
2. Type: Tests. Date: 5/07/84
3. Time: 15.32.41.10
4. Name
5. Given time
6. Output time of problem
7. Completion interval
8. Data register
9. Processor number
10. Priority
11. Completion time
12. Characteristics
13. Conformity

An analysis of problem distribution by processors, their time and period of completion, based on the record obtained, makes it possible to come to some conclusions about the distribution of computer resources.

The method of issuing a special command may be different in each actual case. The choice is determined by the architectural features of the computer complex, and also by the method of connecting the monitor unit directly to the processors or through the input-output interface. This may be commands to transfer information or input-output commands when the monitoring unit is connected as a peripheral device, as well as supplementary commands, introduced into the computer architecture and having its code, but not requiring any kind of operations in the processor. The determining property of such a command should be the possibility of transferring the main parameters of the problem and its beginning and end signals to the monitoring unit.

Thus, in the functioning process of programming for a realtime system (during comprehensive debugging, tests or use) the next problem having received control, transmits to the monitoring unit information on its startup or ending. The device applies this·information to one of the input registers, the information from which further enters through a multiplexer to the data register (Figure 3). Certain stages of the register determine the number of the processor from which the command originated. The multiplexer in turn switches its inputs to the input registers, that is, operates in a cyclical mode with the connection period sufficient to pick up the path of the data record in the buffer memory given the simultaneous arrival of commands from several processors.

The control circuit ensures the decoding of arriving commands and controls the compliance of the record to the buffer memory, consisting of two identical units, using a multiplexer. In the memory the current time, arriving from a time counter, is recorded also. Consequently, each record contains the processor number, the start time of the problem, the name code and other parameters.

The entry counter has a capacity equal to the buffer memory capacity, and establishes a readress record. Given an overfilling of the counter a switch over of the multiplexer to the record in the second memory unit occurs and a signal is issued to the control unit according to which a playback of the accumulated information from the first memory unit occurs through multiplexer 2.

The control unit processes the derived data programmatically by searching for the remaining parameters of the problem in the reference handbook by name, such as priority given time, completion interval, and determines the completion time, etc. Then, a record is printed showing the correspondence of the given parameters to be valid for the next analysis.

Analogously, the examined approach makes it possible to register both internal data flow and external information. In the process of using programs such a monitoring principle permits the operational system to be observed continuously, and in the case of outages or failures accumulated diagnostic information will make it possible to conduct a purposeful search for the cause of the error.

Thus, the example under examination, of implementing the suggested approach, attests to the fact that a new organizational principle for monitoring the program functioning of a real-time system will make it possible to obtain information on the operation of a complex of programs in the system with minimal interference in its operation, without influencing the natural progress of program completion and without destroying real-time interrelationships, which is particularly important for multiprocessor control systems functioning in real time. Such a principle makes it rather simple to broaden the available possibilities, to complete the selection of parameters of interest and introduce new functions, thanks to the combination of hardware and software.

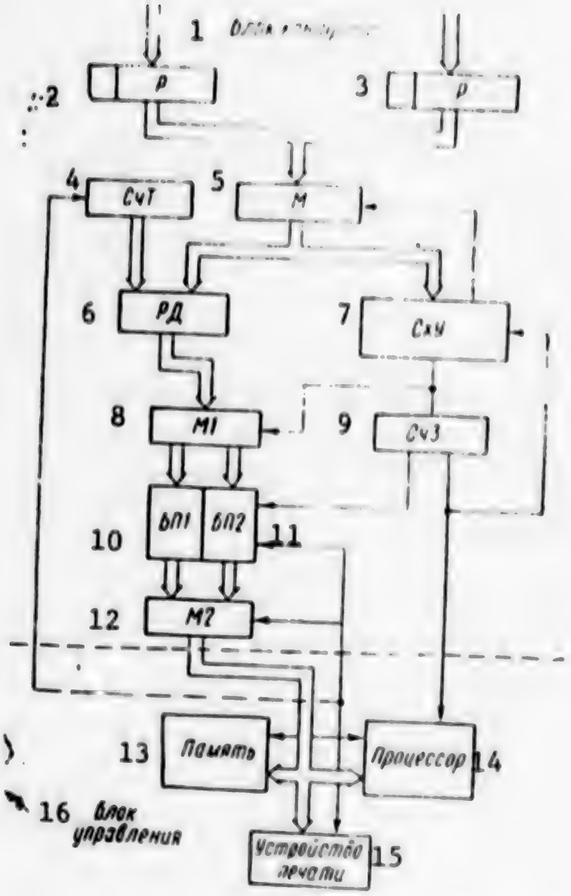


Figure 3. Functional Schematic of Control Device

Key:

1.	Monitoring unit	9.	Entry counter
2.	Register	10.	Buffer memory 1
3.	Register	11.	Buffer memory 2
4.	Time counter	12.	Multiplexer 1
5.	Multiplexer	13.	Memory
6.	Data register	14.	Processor
7.	Control circuit	15.	Printing device
8.	Multiplexer 1	16.	Control unit

The information obtained during the monitoring process makes it possible to determine the degree of load, both for the system as a whole and for each of its processors, both in time and in informational capacity, which makes it possible to finish debugging and testing the real-time system in a shorter time period, and, in the same manner, reduces the costs of developing, introducing and supporting programs for complex systems.

Summary

An efficient organization of software control in complex computation systems is possible on the basis of special hardware-software means which do not disturb operation of the object under control.

Literature

1. Boehm, B.W., "Software and Its Impact: a Quantitative Assesment," DATAMATION, 19, No. 5.
2. Boehm, B., Braun, Dzh., Kaspar, Kh., "Kharakteristiki kachestva programmnogo obespecheniya" [Software Characteristics], Moscow, Mir, 1981, 208 pp.
3. "Agregatnaya sistema programmnogo obespecheniya. Sistemnyye obrabatyvayushchiye programmy. Rukovodstvo po pol'zovaniyu" [Aggregate Programming System, Systems Processing Programs. Instruction Manual], Severodonetsk, Rotaprint NPO "Impul's," 1977.
4. "Programmnoye obespecheniye SM-3. Programma OTLADCHIK" [Programming the SM-3. The Debugger Program], Moscow, In-t elektronnykh upravlyayushchikh mashin, 1977.
5. Faradzhev, V. A., Sigalov, I. L., Zvarich, S. I., Finkel'shteyn, V. A., "Complex Computer Hardware for Debugging Real-Time Programs," UPRAVLENIYE SISTEMY I MASHINY, 1980, No. 1, PP. 49-55.
6. Grishchenko, A. Z., Grishchuk, V. P., Denisenko, V. M., "Avtomatizirovannyye sistemy upravleniya tekhnologicheskimi protsessami" [Automatic Systems for Controlling Production Processes], UkSSR Academy of Sciences Academician B. B. Timofeyev (editor), Kiev, Tekhnika, 1983, 350 pp.
7. Lipayev, V. V., "Nadezhnost' programmnogo obespecheniya ASU" [Program Reliability of Automated Control Systems], Moscow, Energoizdat, 1981, 240 pp.

12318
CSO: 1863/100

UDC 681.3.06+65.001.56

GRAPHIC FORM FOR WRITING ALGORITHMS AND PROGRAMS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 84 (manuscript received 17 Jul 84) pp 51-56

[Article by I. V. Vel'bitskiy, L. D. Raykov, A. V. Maklakov, A. A. Mkrtumyan and I. B. Ushakov]

[Text] Introduction. The characterization of software as an industrial product makes the task of analyzing the symbolic systems, which are now used for writing algorithms and programs, an urgent one. There are now two such systems, and neither of them satisfy the requirements for the industrial organization of programming work.

The textual form for writing algorithms and programs as formulas and operators in some artificial programming language is the more widely used one. This form is convenient for presentation on modern alphanumeric input-output devices, but it is not visual, does not reflect the structure of a program as an item, and uses the powerful visual apparatus and associative thinking of man very nonproductively. The use in it of numerous key words of the type IF, FOR, CASE and others, comments, staggered printing, block brackets of the BEGIN-END type and square brackets for vertical coupling between elements of a language makes the corresponding symbol system's alphabet even more complicated and does not contribute to a fundamental solution of the problem.

The graphic form that is now used to write algorithms and programs in the form of flow charts that are notated at their vertices (block diagrams, SDL diagrams, the information, logic and flow plans of the ISO standards and others) is visual, but it has one very substantial flaw: the impossibility (ineffectiveness, technological nonfeasibility) of representation on modern alphanumeric input-output devices. The existing graphic form for writhing algorithms and programs cannot be kept in a computer, and all attempts to make this an automatic process have so far met with a lack of success. The production of drawings (block diagrams) of programmed items by hand (at a drafting table) is impossible, because their logical complexity far exceeds everything that man has made by industrial methods up until now. Therefore, work on existing symbol systems without the appropriate continuous (at all stages, from the beginning to the end of the development of a program) automation means is impossible.

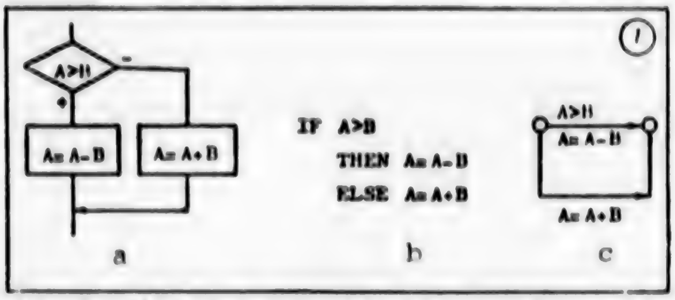


Figure 1. Example of writing a conditional statement in various symbol systems: a. in the form of a block diagram; b. in textual form; c. in the form of an R-scheme.

In this article we propose a new graphic form (symbol system) for writing algorithms and programs that is free from the flaws enumerated above. The new form utilizes flow charts, called R-schemes, that are notated along the arcs (Figure 1). In such flow charts, the condition (logic expression) of passage along the arc is written above it, and the action (operator or uninterrupted sequence of several operators) performed in connection with it is written beneath the arc . In the new symbol system it is possible to write any program in any known programming language; in comparison with the traditional textual symbol system, the notation that is produced can be entered in computer in one-third to two-thirds the time, occupies only half the amount of memory, and requires an average of five-ninths less paper.

The main advantage of the new symbol system is the possibility of creating continuous automation of the programmer's work and a visual representation of not only algorithms and programs, but also data, information and logic circuits, SDL diagrams, network schedules, technological routings and so on.

At the present time a standard, "Unified System of Programming Documentation: Algorithm and Program R-Schemes--Conventional Graphic Notation and Implementation Rules," is being formulated for this symbol system. The standard establishes the conventional graphic notation for the elements and structures of R-schemes, as well as rules for their realization automatically (with the help of a computer) and/or manually. R-schemes can be used for the depiction of algorithms and programs in all programming documents defined by GOST [All-Union State Standard] 19.101-77.

The new standard does not exclude the use of previously adopted standards (GOST 19.002-80 and GOST 19.003-80) for the depiction of algorithms and programs by the traditional method, in the form of flow charts notated at their vertices. The choice of the standard for the realization of a specific programming document is determined by the corresponding technical assignment in this case.

Machine support of the new writing form makes it possible to combine the programming documents "Program Text" and "Program Description" into one document ("Program Text"). In this case, all the information that must be in the "Program Description" document (according to GOST 19.101-77) is entered in "Program Text": information about the logic structure and functioning of the program that is obtained during the process of its development. In addition, in the technical assignment for the development of this programming item, it must be indicated explicitly that it will be documented by means of R-schemes in the "Program Text" containing the information mentioned above. Such a capability reduces the laboriousness of the formulation of the programming documentation when the new standard is used and improves its quality, since the "Program Text" document reflects the program development process and automatically carries out all the changes in the process of program debugging and utilization.

The basic concept of the new symbol system is the concept of the R-scheme, which is understood to mean a structured graph that is notated along the ribs. A structured graph is an oriented graph consisting of structures (subsidiary graphs), each of which has only one input and one output. Each R-scheme can be accompanied by a text that is placed ahead of and/or behind it. The R-scheme text is yet another important concept in the new symbol system. An R-scheme text is any text, including an empty one (that is, it is not mandatory), a formula, a formal text (in programming languages) and any special symbols, tables, figures and so forth. The absence of limitations on an R-scheme text and its formulation make the proposed symbol system a general-purpose one and enables it to be used in different areas.

Elements of R-schemes. The elementary concepts of R-schemes include the vertex, the special vertex, the arc, the special arc, the connecting line and the comment. The conventional graphic notations for these elements can be realized by hand or in an automated fashion, on graphic or standard alphanumeric computer input-output devices.

A vertex of an R-scheme is represented by a circle at least 2 mm in diameter, or by the symbol "+" during realization on alphanumeric input-output devices. A vertex designates the beginning and/or end of an R-scheme's structure.

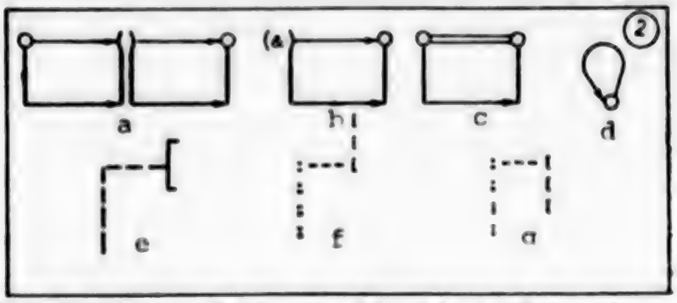


Figure 2. Examples of the realization of R-scheme elements.

The special vertex is introduced by the standard in order to distinguish an R-scheme structure for the purpose of a visual and unambiguous depiction of it or for the identification of a special realization. It is designated by parentheses (the left and right parentheses are at a distance of at least 1 mm from each other). The special vertex is used for the separate depiction of the vertical connecting lines of two side-by-side structures (Figure 2a), as well as for the identification of a special realization of the structure by the entry of a single line of R-scheme text between the parentheses (Figure

2b), where "&" identifies the type of special structure, which is interpreted as two retained commands that are executed in parallel [2].

The arc and the special arc are also R-scheme elements. A arc is oriented and is depicted by a horizontal line with an arrow pointing to the left or right. A arc designates a transition from one R-scheme vertex to another. When R-schemes are being realized on alphanumeric input-output devices, the arc is depicted by a chain of "-" (minus) symbols and ">" (more than) or "<" (less than) symbols at the ends of the rib.

A special arc is depicted by two unoriented horizontal lines that are at least 1 mm apart (see Figure 2c). On alphanumeric input-output devices, it is indicated by a chain of "=" (equals) signs. The special arc also designates a union of two R-scheme vertices, the function of which is determined by the specific realization. As a rule, if nothing special is stipulated for a realization, "=" symbols or two parallel lines between vertices mean the identification of these vertices as one in such a fashion that the other (nonspecial) arcs around these vertices form a loop-type flow chart. For example, the flow chart in Figure 2c, which is depicted in accordance with the standard, is equivalent to the flow chart in Figure 2d, which is in the traditional (nontechnological) notation system. These graphs are used primarily for the depiction of a cycle of the WHILE type.

In those cases where two vertices are connected by more than one arc, a vertical connecting line is used. On alphanumeric input-output devices, the connecting line is represented by "!" symbols arranged one under the other.

The comment R-scheme element (see Figure 2e) is used to identify a coupling between R-scheme elements and their appropriate explanatory text. A comment is depicted by a vertical (upward or downward) and a horizontal (to the left or right) dotted line that ends in a square bracket, behind which the explanatory R-scheme text follows. For realization on alphanumeric input-output devices, a comment is depicted by the symbols ":" (vertical line) and "-" (horizontal line and a square bracket (for each line of explanatory text). In this case the horizontal line leads to the first or last square bracket (see Figure 2e, 2f).

R-Scheme Structures. The standard establishes two structures: basic and special. A basic structure is defined as two vertices connected by one or more arcs oriented in any direction (left to right or right to left) and in any combination (arc succession from top to bottom is arbitrary). Examples of the depiction of basic structures are presented in Figures 3a and 3b for manual and 3c and 3d for automatic realization methods. A basic structure designates a sequence of transitions between vertices in accordance with the directions of the arcs.

A special structure is defined by the standard as two vertices connected by a special arc and any number of arcs in any combination. Examples of the realization of special structures are presented in Figures 4a and 4b for manual and 4c and 4d for automatic production methods. Arcs may be missing (see Figure 4b). A special structure designates a sequence of transitions between

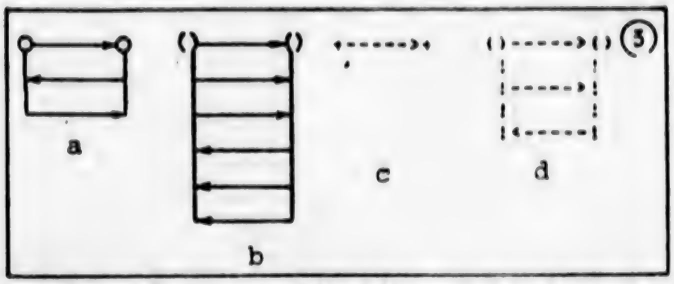


Figure 3. Examples of the realization of R-scheme basic structures.

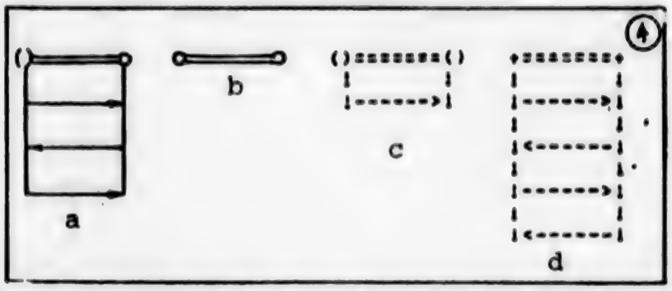


Figure 4. Examples of the realization of special structures.

vertices that are realized in a special manner determined by the specific realization.

The structures (basic and special) defined by the standard have one input and one output. The initial and final vertices of these structures are determined in accordance with this. The structure vertex from which the first upward rib originates is called the final one. For a special structure that does not contain ribs (see Figure 4b), the initial and final vertices are determined by its specific utilization (location in the R-scheme).

In addition to these two structures (basic and special), the standard allows any structures derived from them by the use of the specific rules for connecting them.

Structure Connection Rules. For two original structures, the standard defines three connection operations: sequential, parallel and enclosed. As a result of the connection of structures, R-scheme structures are obtained that, in turn, can participate in all the indicated operations. This makes it possible, with the help of the connection rules established by the standard, to obtain structures of arbitrary complexity, with the resulting structure always having one input (the initial vertex) and one output (the final vertex).

A sequential connection of R-scheme structures is realized by merging the final vertex of one structure with the initial vertex of another structure that is placed behind it (Figure 5). When the vertices are merged, the structures' connecting lines are depicted separately, by a special vertex. The initial vertex of the first and the final vertex of the second of the structures being joined become, respectively, the resulting structure's intial and final vertices.

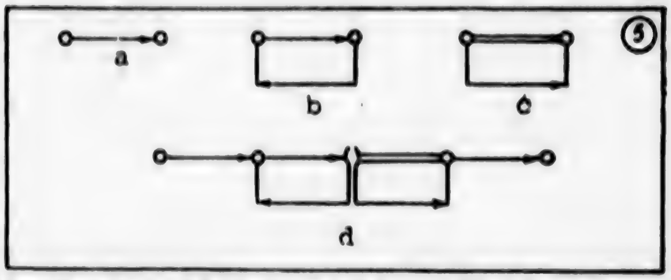


Figure 5. Result of sequential connection (d) of structures a, b, c and a.

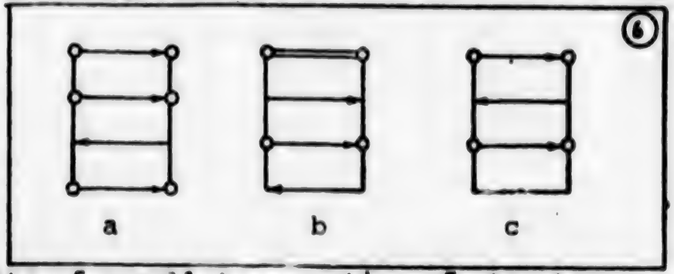


Figure 6. Results of parallel connection of structures: a. of those depicted in Figures 5a, 5b and 5a; b. of those depicted in Figures 5c and 5b; c. of those depicted in Figure 5b.

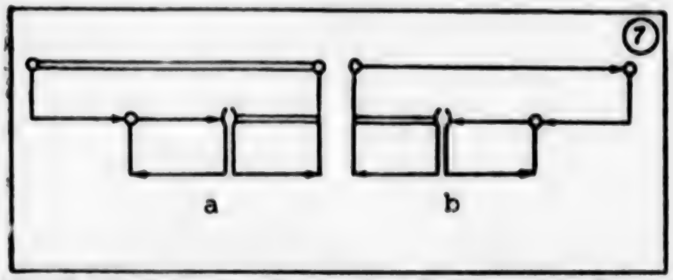


Figure 7. Results of enclosure of sequential connection of structures: a. of those depicted in Figures 5b and 5c in the structure depicted in Figure 5c; b. of those depicted in Figures 5b and 5c in the structure depicted in Figure 5b.

A parallel connection of R-scheme structures is realized by using vertical lines to connect the initial and final vertices of one structure to the initial and final vertices, respectively, of another structure that is below it. The initial and final vertices of the first (upper) of the connected structures become, respectively, the initial and final vertices of the resulting structure. Examples of the realization of the parallel connection of structures are presented in Figure 6.

An enclosed connection of R-scheme structures is realized by replacing a rib of the structure in which the enclosure is being made with a sequential connection of a rib, oriented in the same direction, of the structure being enclosed. In connection with this, the connecting line from the final vertex of the enclosed structure and the final vertex itself (if it is not the final vertex of the structures' parallel connection) merge, respectively, with the connecting line and vertex to which the replaced rib was connected. The initial and final vertices of the resulting structure remain, respectively, the initial and final vertices of the structure in which the enclosure is being made. There can be any depth of enclosure.

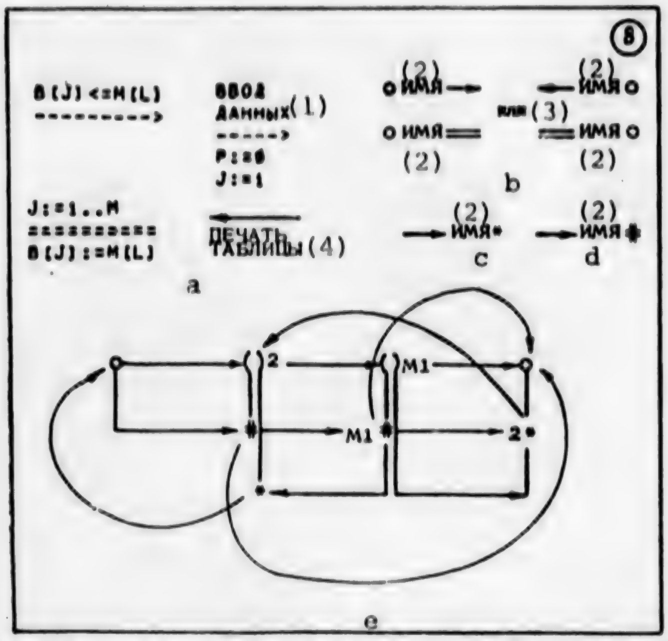


Figure 8. Examples of realization of superscripts on R-scheme elements and structures.

Key:

1. Data input	3. Or
2. Name	4. Print tables

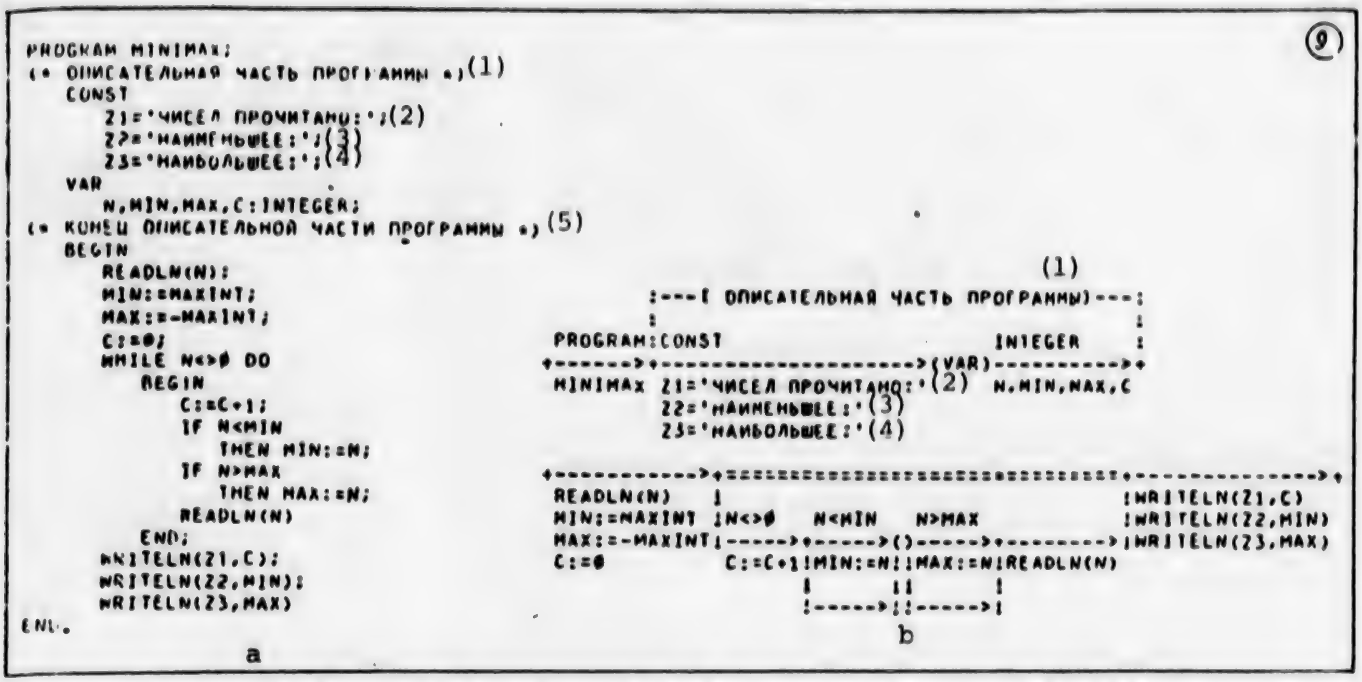
Examples of an enclosed connection of structures for ribs going in different directions are presented in Figure 7.

Arbitrary R-schemes can be constructed with the help of the structure connection rules defined above. These rules are simple and graphic and make it possible to improve one's understanding of written algorithms, in addition to making the logic of the programs transparent.

Superscripts on R-Scheme Elements and Structure. In order to write algorithms and programs with the help of R-schemes, the elements and structures are notated with texts. For this purpose, the standard establishes the appropriate superscripts for R-scheme elements and structures.

A superscript inside a special vertex (any single line of R-scheme text) gives the type of structure and designates its special realization.

Loading of the arcs (and special arcs) is the characteristic feature of the new symbol system. In accordance with this, the standard establishes superscripts above (and beneath) arcs that are one (or more) line(s) of text, as stipulated above, that is/are written in such a fashion that the length of any line does not exceed the length of the corresponding arc (or special arc). Superscripts above (and beneath) a arc define the condition for passage along the arc and the action performed during passage along the arc, respectively, and for a special arc determine a special realization of the entire R-scheme structure (such as one that is determined during the realization). It should be mentioned that superscripts on arcs are not mandatory and can be absent. Examples are presented in Figure 8a.



```
PROGRAM MINIMAX;
(* ОПИСАТЕЛЬНАЯ ЧАСТЬ ПРОГРАММЫ *)(1)
   CONST
      Z1='ЧИСЕЛ ПРОЧИТАНО:';(2)
      Z2='НАИМЕНЬШЕЕ:';(3)
      Z3='НАИБОЛЬШЕЕ:';(4)
   VAR
      N,MIN,MAX,C:INTEGER;
(* КОНЕЦ ОПИСАТЕЛЬНОЙ ЧАСТИ ПРОГРАММЫ *)(5)
   BEGIN
      READLN(N);
      MIN:=MAXINT;
      MAX:=-MAXINT;
      C:=0;
      WHILE N<>0 DO
         BEGIN
            C:=C+1;
            IF N<MIN
               THEN MIN:=N;
            IF N>MAX
               THEN MAX:=N;
            READLN(N)
         END;
      WRITELN(Z1,C);
      WRITELN(Z2,MIN);
      WRITELN(Z3,MAX)
END.
```

```
                                                    (1)
        :---( ОПИСАТЕЛЬНАЯ ЧАСТЬ ПРОГРАММЫ)---:
        :                                     :
PROGRAM:CONST                      INTEGER    :
+------>+---------------------->(VAR)---------->+
MINIMAX Z1='ЧИСЕЛ ПРОЧИТАНО:'(2)  N,MIN,MAX,C
        Z2='НАИМЕНЬШЕЕ:'(3)
        Z3='НАИБОЛЬШЕЕ:'(4)

+----------->+=============================+-------------->+
READLN(N)    !                              !WRITELN(Z1,C)
MIN:=MAXINT  !N<>0    N<MIN    N>MAX        !WRITELN(Z2,MIN)
MAX:=-MAXINT !----->+------>()----->+------>!WRITELN(Z3,MAX)
C:=0          C:=C+1!MIN:=N!!MAX:=N!READLN(N)
                    !      !!      !
                    !----->!!----->!
```

Figure 9. Example of realization of R-scheme of program in PASCAL.

Key:

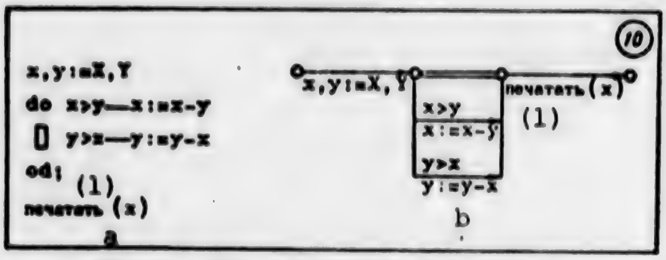
1. Descriptive part of program
2. Quantity of numbers read
3. Least
4. Most
5. End of descriptive part of program

Any R-scheme structure can be labeled with a name that is written near the structure's initial vertex without spacing (see Figure 8b), where NAME is an identifier or a sequence of decimal numbers.

The name of a structure makes it possible to organize error diagnostics in R-schemes, as well as structurized transitions in them that are realized with the help of structurizers. The standard defines a transition structurizer to the beginning (see Figure 8c) and a transition structurizer to the end (see Figure 8d), where NAME with the symbols "*" or "#" is written at the end of the rib without a gap. The symbols "*" and "#" are on the level of a vertical connecting line, and NAME can be absent.

A structurizer to the beginning or end designates, respectively, a transition to the beginning or end of the R-scheme with the indicated name. If the name is absent, the transition is made to the beginning or end, respectively, of the given R-scheme (see Figure 8e).

R-Scheme Realization Rules. R-schemes are a convenient form for writing algorithms and programs, so they can be used in different programming documents formulated in accordance with the YeSPD [Unified System of Programming Documentation]. In programming documents, R-schemes are written on Form #1 or #2 sheets (GOST 19.106-78). On each sheet it is possible to have one or more R-schemes, together with accompanying text. In connection with this, R-schemes can be either drawn up in the form of separate illustrations (drawings)



```
x,y:=X,Y
do x>y—x:=x-y
[] y>x—y:=y-x
od;  (1)
печатать (x)
```

Figure 10. Euclid's algorithm.

Key:
1. Print

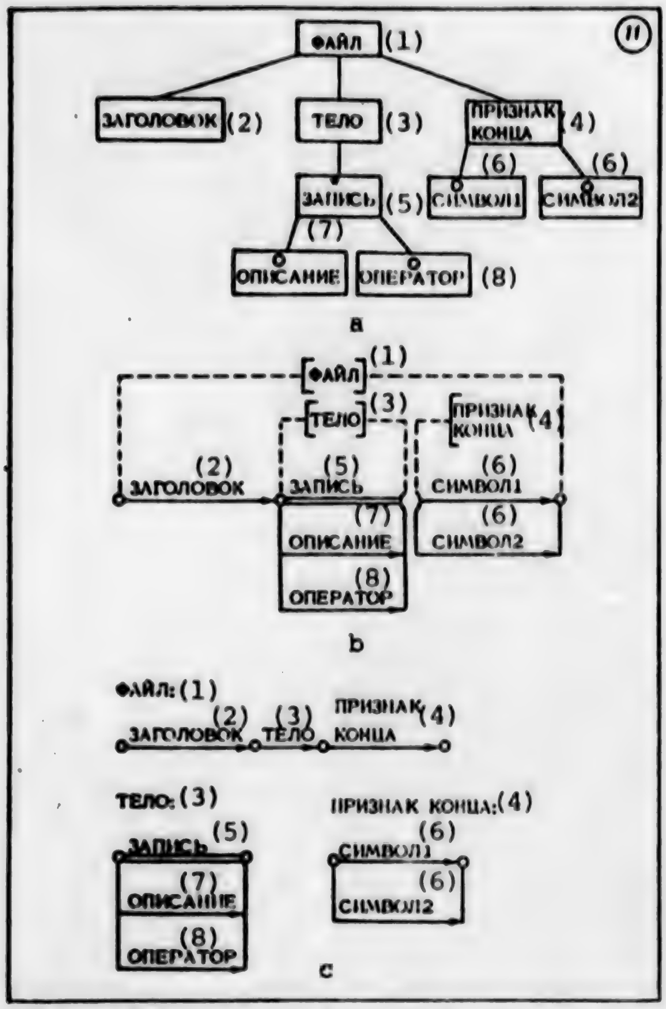


Figure 11. Description of data structure.

Key:

1. File	5. Write
2. Heading	6. Symbol .
3. Body	7. Description
4. End tag	8. Operator

or supplements, or placed in the gap between lines of document text without the corresponding numbering of the R-schemes as figures. This makes it possible to realize compact documentation of the programmed text with the use of

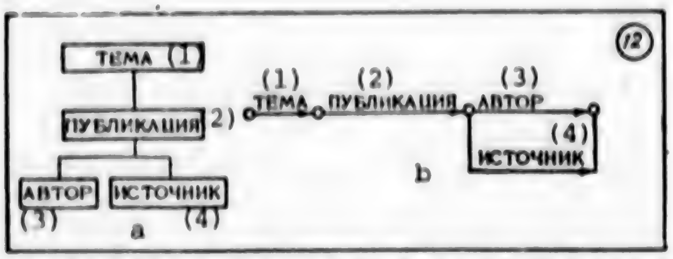


Figure 12. Hierarchical structure of a data base for publications.

Key:
1. Theme
2. Publication
3. Author
4. Source

R-schemes. The programming document text accompanying the R-scheme text and the R-scheme itself are separated from each other by intervals that are sufficiently large for them to be perceived separately.

The use of R-schemes in programming documents makes it possible to improve the quality of programming documentation, reduce the amount of labor required for its production, and reduce the number of instructions given the user. In connection with this, there is a significant increase in the level of automation involved in the preparation of programming documents.

Let us present an example of the writing of algorithms and programs in the new symbol system. A MINIMAX program analyzes a series of whole numbers and prints the quantity of numbers read and the minimum and maximum numbers. The symbol for the end of a series of numbers is zero. In the example we present the traditional linear (Figure 9a) and graphic (Figure 9b) forms for writing a program in PASCAL. The graphic form is represented by two R-schemes: the first corresponds to the descriptive part of the linear program and the second to the execution part.

As an example of writing an algorithm in the new symbol system, we will present the well-known algorithm for finding the largest common denominator of two arbitrary natural numbers X and Y. Figure 10a shows this algorithm written in linear form, as retained Deyks"ra commands [2], whereas Figure 10b shows it written in the proposed symbol svstem.

Figure 11a shows the writing of a data structure in Jackson diagrams [3], where the symbol "*" designates a relationship of the "repeat" type and "O" means "selection from." Two equivalent writings of this same data structure in R-schemes are shown in Figures 11b and 11c.

Figure 12 depicts a hierarchical structure of a physical data base that contains information about publications concerning a series of selected themes. Figure 12a is this data base structure written in traditional form, and in Figure 12b it is written in R-schemes.

A syntactic description of a list of identifiers, using (Backus's) metalinguistic formulas, is presented in Figure 13a, whereas Figures 13b and 13c are corresponding equivalent descriptions utilizing R-schemes.

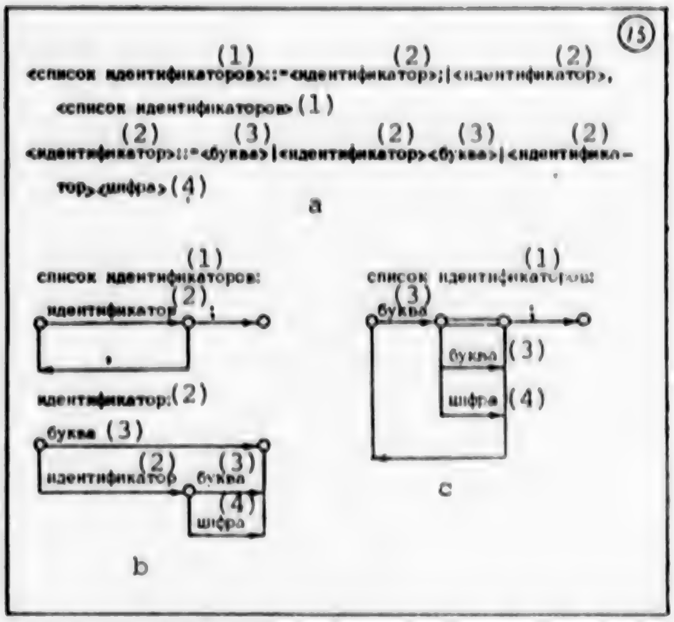


Figure 13. Synthetic [sic] description of an identifier.

Key:

1. List of identifiers	3. Letter
2. Identifier	4. Number

Conclusion. The new symbol system has automatic support from all of the main Soviet computers: YeS EVM [Unified System of Electronic Computers], SM EVM [International System of Small Computers], microcomputers of the "Elektronika-60" type and DVK-2 personal computers. The following widely used programming languages have been loaded into the graphic symbol system: PL-1, FORTRAN, RTRAN, Assembler (YeS EVM); FORTRAN, PASCAL, RTRAN, Assembler (SM EVM); PASCAL, FORTRAN, Assembler (microcomputers).

The introduction of the standard for the new graphic system for writing algorithms and programs will contribute to an improvement in the quality and mobility of the programming product and reduce the amount of labor required to develop it and prepare design documentation for software which is an industrial product.

BIBLIOGRAPHY

1. Yershov, A.P., "Programming Lexicon," in "Avtomatizatsiya proizvodstva paketov prikladnykh programm i translyatorov: Tez. dokl. 2-y Vsesoyuz. konf." [Automating the Production of Packages of Applied Programs and Translators: Summaries of Reports Given at the Second All-Union Conference], Tallin, 1983, pp 3-5.

2. Deykstra, E., "Distsiplina programmirovaniya" [Programming Discipline], Moscow, Izdatel'stvo "Mir", 1976, 276 pp.

3. Jackson, M.A., "Principles of Program Design," London, Academic Press, 1977, 299 pp.

4. "Yedinaya sistema programmnoy dokumentatsii" [Unified System of Programming Documentation], Moscow, Izdatel'stvo standartov, 1982, 128 pp.

5. Vel'bitskiy, I.V., "Paperless Technology for Programming in a Dialog Environment," UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, 1982, pp 29-37.

11746
CSO: 1863/152

UDC 683.3.06

CROSS-ASSEMBLER SOFTWARE FOR MICROPROCESSOR SYSTEMS

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 27 Dec 83) pp 62-66

SHCHERBAKOVA, N. G., Novosibirsk

[Abstract] Three cross-assemblers and two inverse cross-assemblers for various microprocessors are described, each matched to a "Nord-10" service computer with the SINTRAN-III operating system and a screen editor of any data text. The cross-assembler CA6800 written in "Nord-PL" prepares absolute object codes for programs written in Assembler languages for MC6800 microprocessors. The cross-assembler CA8708 written in PASCAL prepares absolute codes for programs written in Assembler language for MCS-48 microprocessors. The cross-assembler CA3000 is a translator from XMAS and a translation into XMAR, with an intermediate file in the computer for programming of microprocessors built with series K589 microcircuits, the XMAS language being expandable and not rigidly fixed except for CPU and JMP functions as well as a few control functions. The inverse cross-assemblers DA6800 and DA8080 restore original texts from object codes for MC6800 and KR580 microprocessors respectively. These inverse cross-assemblers are dialog programs, inverse assembly consisting of three successive processes: separation of instructions from data, establishment of data format and marker spacing, and restoration of original text. The algorithm used for the first of these processes ensures a complete solution of the problem, despite inevitable ambiguities in instruction decoding, provided that the original program contains no codes which modify instructions, that transfer addresses are given explicitly without conditional branching where unconditional transfer takes place, and that reentry from a subroutine occurs at a point which follows the corresponding instruction. All this software is a part of a distributed system for development of microprocessor hardware and software. The author thanks P. M. Peslyak for discussing each step, also V. A. Meleshikhin and Ye. G. Yurashanskiy for helpful suggestions. Figures 1; references 9: 3 Russian, 6 Western. [74-2415]

APPLICATIONS

'INTELLEKT' COMPUTER SYSTEM SERVES DIVERSE USERS

Yerevan KOMMUNIST in Russian 14 Nov 84 p 1

[Article by G. Novikov, Minsk: "Science for the Five-Year Plan Period; 'Intellekt' on the Line"]

[Text] Specialists of the USSR Goskomizdat [State Committee on Matters of Publishing Houses, Printing and the Book Trade] obtained answers practically instantaneously to the most complex questions relating to publishing activities. However, for a week they did not turn for information to the data processing and computing center of their own department, but to a computer located in Minsk. The task proved to be just as simple as any telephone conversation.

Now it is sufficient for Muscovites to dial through the automatic long-distance telephone exchange a Minsk number belonging to the "Intellekt" computer system of the Belorussian SSR Minzhilkommunkhoz [Ministry of Public Housing] and communication is established between a human being and a computer. Their dialogue takes place not in the language of numbers and characters customary in such situations, but in ordinary words and sentences.

Whether a specific book has a reader demand, whether it is feasible to reissue it, and how large a run is required to cover expenses--the computer gives a spoken answer to dozens of similar questions arising in the daily work of the USSR Goskomizdat. On request, it analyzes the work of individual publishers and the industry as a whole, makes an economic forecast, suggests what decisions to take in a specific situation, and recommends how to straighten things out.

"The 'Intellekt' system has already become a reliable adviser for leading specialists of our ministry," tells E. Medvedev, the director of the special design and technology bureau for control systems of the Belorussian SSR Ministry of Public Housing. "Not a single serious decision is taken by the sector's leadership until the computer analyzes the situation and gives its forecast."

"Having handed over our system for the duration for use by specialists of a department quite remote from the problems of urban management and located in another city, we set the important objective--without freeing the 'Intellekt' from its main job--of teaching it the parallel solution of problems of another nature."

"In a total of 10 days the programmers of the ministry and Goskomizdat 'reformed' the system's logical thinking. This cooperation turned out to be mutually advantageous. The cost of long-distance telephone conversations between Moscow and Minsk is insignificant in comparison with the cost of acquiring one's own computers and speech analyzers and synthesizers and of developing software and servicing computers."

8831
CSO: 1863/126

GRAPHICS READING DEVICE OF 'INTELLEKT' SYSTEM DESCRIBED

Moscow IZVESTIYA in Russian 7 Jan 85 p 2

[Article by G. Novikov, Minsk: "Alongside the Researcher; 'Eyes' of Talking Computer; We Talk About One Research Study Which Is Expanding the Capabilities of Today's Computers"]

[Text] Unlike series-produced computers, the computers of the special design and technology bureau of control systems of the Belorussian SSR Minzhilkommunkhoz [Ministry of Public Housing] are capable not only of solving the most complicated problems in planning and economic forecasting, but also of hearing and understanding human speech, speaking in a generally easily understood language, and seeing objects of any configuration and size. However, this too seemed insufficient to the developers of the unique system—they are attempting to furnish the computer with "eyes" by means of which the computer will read drawings and other graphic information. Having implemented what they have planned, the specialists will contribute an appreciable share to the development of the next-generation computer.

In the room where the "Intellekt" data processing and control system was located, the atmosphere is outwardly the same as in the majority of computer centers--several large YeS-1035 and YeS-1022 computers and, interfaced with them, "Elektronika 100-25" minicomputers, microcomputers and voice synthesizers and analyzers. A single facility seems to be here by chance--a drawing board with a straight edge attached to one of its edges and a device somewhat reminiscent of a medical phonendoscope. This likeness of a designer's Kuhlman drafting unit is connected to computers by means of two cables. I turned for a commentary to the developers of the system, candidates of technical sciences Bureau Director E.M. Medvedev and E.K. Skvortsov, chief engineer of the Novosibirsk Institute of Thermal Physics of the USSR Academy of Sciences.

"We are interfacing Skvortsov's instrument--a semiautomatic device for copying graphic information--with the 'Intellekt,'" E.M. Medvedev explains the essence of what is going on. "Said more simply, we are teaching a computer to take a mathematical copy from a drawing and analyze and remember what was 'seen.'"

The researchers line up a small sighting tube with one of the lines on a graph, press a button and immediately a light signal appears on a monitoring oscilloscope. After an instant, bright green digits flare up on the display's screen, a printer chirps, and the computer communicates to the operator the first coordinate of one of the lines.

"The ultrasonic graphic image reader interfaced with the 'Intellekt' is relatively simple," the inventor of the device, E. Skvortsov, relates. "Three miniature receivers are mounted in a special straight edge fastened to a plotting board. And there is a transmitter in the sighting device with which we outline the drawing's lines. The receivers sensitively register the slightest displacement and find its direction, and all this information is converted into codes which enter the computer. The method is fairly precise--distortions do not exceed 0.1 mm; not a single pen writes as finely."

The new device has one more advantage of no small importance--it is 20 times less expensive than existing electromagnetic analogues, although original and of a quite simple technical design. Judge for yourself: All the parts making up the device are series produced and the sample tested in Minsk the engineer assembled right there, in the SPTB [special design and technology bureau].

"We are teaching our system to recognize and analyze drawn images not for the sake of an experiment," Eduard Mikhaylovich continues. "This problem has disturbed scientists and designers almost as long as computers have existed--after all, nearly all research instruments output information in the form of all kinds of graphs, and computers, on the other hand, have 'become accustomed' to deal only with digital data. The computer's 'power to discern' will eliminate the disharmony, i.e., its ability to rapidly and, the main thing, accurately translate into its own machine language drawings, figures, oscillograms, x-rays and graphics in general. This without doubt will truly open up unlimited possibilities in many fields of human activity."

8831
CSO: 1863/150

ROBOTICS PIONEER WANTS INDUSTRIES TO SHARE AUTOMATED PLANTS

Moscow PRAVDA in Russian 13 Jan 85 p 2

[Article by M. Vasin]

[Abstract] The lengthy article reports on an idea proposed by Prof M. B. Ignat'yev which calls for the creation of regional, inter-industry flexible automated production facilities, as an economy measure. The article points up the irony of the fact that Ignat'yev, a pioneer in robotics, was an early advocate of the push for robotization of manufacturing, and now encounters opposition from industrialists in his proposals for a cautious, economy-minded approach to robotization.

Ignat'yev, a professor of the Leningrad Institute of Aviation Instrument Building and a State Prize laureate, is identified as the founder and head of one of the country's first robotics laboratories, and of the first laboratory of multiprocessor computer structures. It is recalled that 20 years ago, having experience in work with robot-manipulators in the nuclear industry, he campaigned for development and widespread production of robots. He is credited with the development of the first Soviet computer-controlled underwater robot. In the 1970's, he is said to have advanced new ideas in computer development. It is recalled that in 1974 at an international congress in Stockholm, he read a paper by a group of Soviet scientists which included V. M. Glushkov and V. A. Myasnikov, on the topic "Recursive Machines and Computer Technology."

It is said that in recent years, having looked closely at the practical results from the first robots in industry, Ignat'yev has become more conscious of the economics of robotization, realizing that the fate of robots depends on how advantageous they are. In the late 1970's, he directed work on one of the first flexible automated production facilities (GAP). With 4 years of experience amassed in the operation of the first GAP's, it is said that Ignat'yev remains an advocate of these facilities, but not at any price. At his institute, he and colleagues have done modeling of the future operation of several GAP's which are now being built, and found that at times their workload will not exceed 10-12 percent. Consequently he sees a need for curbing overenthusiasm and preventing automation for its own sake. He foresees that in some cases, GAP's that would be cost-prohibitive for any single plant could be economically feasible if they served neighboring plants with similar manufacturing technology. Therefore he proposes the creation of regional flexible automated production centers serving more than one industry.

The article implies that there is opposition to this proposal from industrialists who have been swept into the drive for automation and robotization.

FTD/SNAP
CSO: 1863/184

ENHANCED-FLEXIBILITY ROBOT HAND MODULES INVENTED

Moscow SOTSIALISTICHESKAYA INDUSTRIYA in Russian 30 Dec 84 p 2

[Article by B. Bubchikov, correspondent, press center, Goskomizobreteniy [State Committee on Inventions], Moscow: "Robot According to Order"]

[Text] On the eve of the New Year the All-Union Scientific Research Institute of State Patent Expert Examination of Goskomizobreteniy made an affirmative decision on an application of a group of scientists of the Institute of the Science of Machines imeni A.A. Blagonravov of the USSR Academy of Sciences.

"The most important engineering solution of those recognized as inventions at the end of the year," B. Nazarov, the head of the committee's machine building section relates, "turned out to be not at all ordinary. We know that the human hand is a unique tool able to perform quite diverse operations. Designers are striving to develop for robots mechanical 'hands' which would not be inferior to ours in terms of their capabilities. In robotics the flexibility of 'hands' is specified by a concept such as the number of degrees of freedom. The human hand has 27 of these degrees. In the majority of modern robots there are six to nine. The more of these degrees there are, the more complicated the operations are which our mechanical helpers will be able to perform."

The invention of associates of the Institute of the Science of Machines, Doctor of Technical Sciences A. Koliskor, Graduate Student K. Arzumanyan and Junior Scientific Associate D. Daych, made it possible to make a qualitatively new leap in this direction. An industrial robot "hand" assembled from modules with 18 degrees of freedom has been developed at the institute and is passing tests. There is still no such machine anywhere in the world. The most striking thing is that the "hand" contains a total of about 10 different kinds of parts instead of hundreds in ordinary robots!

A new view of the principles of kinematics and control made it possible for the scientists to develop this "hand." The name of the invention goes like this: coordinate space device. As we know, the position of any object in space is stipulated in a coordinate system. Almost all modern devices making it possible to find out where an airplane, ship, automobile or the working member of a robot or manipulator is at a given moment of time are designed precisely on this basis. Methods of controlling the motion of objects in space are also developed in accordance with a coordinate system.

The inventors suggested a new coordinate system. Without going into the fine points, it is possible to say that its main difference from the present consists in the fact that angular values are totally absent in it and the position of an object in space is specified only by linear values.

The principle of linear coordinates opens up extensive capabilities in the development of control systems with feedback for the end element of a robot--its "tongs." This will make it possible considerably to reduce the weight of robots, to increase their labor productivity, and to simplify systems for controlling mechanical helpers.

[Question] What prospects has this invention opened up for industry?

[Answer] "Without a doubt the prospects are very enticing. But for the time being it is possible to speak only of studies which have been made. Work on practical utilization of the new coordinate system began only a year ago. During this time specialists at the Institute of the Science of Machines developed on its basis a unique 'trajectory-time' diagnostic system. It determines and registers all movements of the 'hand' of a robot in the area of the 'tongs.' This is extremely important for the study and further improvement of mechanical helpers. In order for the robot to 'feel' the moment and force of a movement, the inventors developed sensors which surpass in their parameters the best models in the world."

Seventeen versions of different robots which can be assembled from the modules developed by them are suggested in the inventors' application. In the future, designers will be able to develop, without special work, robotic systems for automating quite diverse production processes.

8831
CSO: 1863/150

UDC 681.324

HIGH-CAPACITY REAL-TIME COMPUTER SYSTEM FOR PROCESSING HYDROPHYSICAL DATA

Novosibirsk AVTOMETRIYA in Russian No 5, Sep-Oct 84 (manuscript received 13 Apr 84, pp 3-12

[Article by I.I. Brodskiy, V.A. Kozlachkov, I.I. Korshever, V.S. L'vov, S.L. Musher, Yu.E. Nesterikhin, S.A. Pavlov, A.A. Predtechenskiy, I.G. Remel', and A.V. Shafarenko: "High-capacity Real-time Computer System for Processing Hydrophysical Data", in the section "Automated Systems for Scientific Research"]

[Text] Introduction. The practice of hydrophysical information has a long tradition. Until recently, experimental hydrophysical systems were built as complexes of specialized analog and digital instruments. Lately, the same kind of measurement complexes have been incorporated into real-time systems controlled by small computers [1].

Simultaneously, computational methods for processing data from hydro- and hydrodynamic experiments have been improved. Besides the traditional processing methods--spectral and correlation analysis--approaches associated with use of regression analysis, adaptive methods of spectral assessment, etc. have been developed [2]. Various algorithms for filtering data have been widely used in studying the problem of the origin of hydrodynamic turbulence; the results of multiple-point measurements of time functions of liquid flow rates made it possible to study the topological properties of attractors in the effective phase space, in particular, to measure fractal and information dimensionality [3]. This program, processed on modern high-capacity computers, has no promise within the framework of analog processing methods traditional in hydrophysics.

The appearance of high-capacity computers in the architecture of real-time systems has made it possible to introduce powerful modern methods of detecting and evaluating signals into the processing of hydrophysical signals. This served as the basis for creating the Hydrophysical Real-time Computer Complex (GFVK) in the Automation and Electronometry Institute, USSR Academy of Sciences Siberian Department [IAiE SO USSR AS], according to the ideology of fourth-generation computer systems.

Problem Statement. One of the variations for using the GFVK that has been achieved is the study of a liquid's transition from a laminar flow to turbulent when velocity is increased, in particular the study of coherent structures in a Couette flow arising at high Reynolds numbers. For this, nearly one hundred hydrodynamic sensors were installed on a hydrodynamic test stand (described in detail in reference [4]), measuring the velocity gradient of a fluid at equidistant points $z = \alpha_n$ along a vertical cylinder. The time-space signal, containing data about the dynamic behavior of the liquid flow, goes from these sensors to the GFVK. Real-time processing of this multi-dimensional flow of data by flexible variable algorithms is the primary task of the GFVK.

The first (and mandatory) stage of data processing in the GFVK consists of parallel computation of the spatial (using sensors) Fourier harmonics of the signal characterizing the various coherent structures in the flow. Following this, it is necessary to compute the power spectrum for time realization of each spatial Fourier component in the simplest processing variant. As a result, this method leads to a two-dimensional Fourier transformation of the initial space-time signal.

The global structure of data in the GFVK is represented in the system as an infinitely developing strip, generated by the system of sensors and produced on an array, whose width represents instantaneous readings of signals coming from the sensors, and whose longitudinal section is time realizations of the signals of each of the sensors. The two-dimensional Fourier transformation of the same signal is done initially by the columns of this rectangular matrix, and then by the rows, representing the endless time sequence of like spatial frequencies.

In the transition from vector operations above the rows to operations above the columns (or vice versa), the two-dimensional array must be transposed. This is not difficult if the entire array is in on-line storage; however, more complex procedures for staged transposition are necessary, using external storage, when there are restrictions on its volume [5].

Because it is necessary to accumulate periodograms [6] of individual overlapping realizations, to increase the signal-to-noise ratio and obtain spectral assessments of signals with satisfactory accuracy, the "endless strip" has a blocked structure. Its individual "frames" overlap with a certain spacing, selected from considerations associated with the achieved variance of the spectral evaluation and allowable accumulation time (Figure 1).

The procedure for obtaining spectral evaluations also includes low-frequency filtration. The two-dimensional spectra are the initial data for analysis of the dynamic properties of the liquid flow. The decision on the coherent structure is made by comparing the levels of the signal statistics and the noise process. Therefore, for dynamic optimization of GFVK operation, it is al;o necessary to assess the statistical characteristics of the turbulent background.

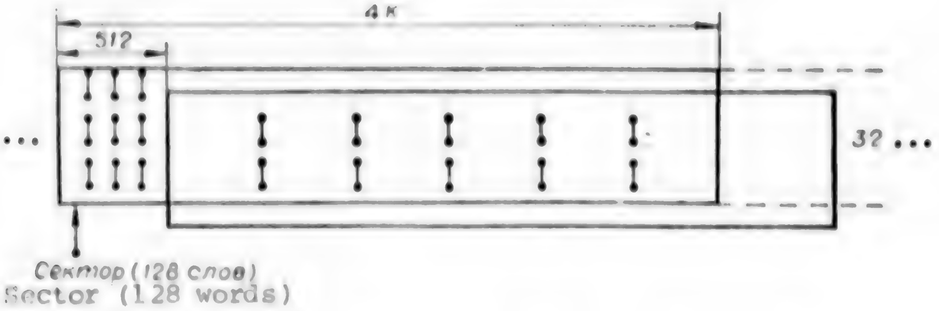


Figure 1. Structure of data in the system

The GFVK is a sufficiently flexible computer complex and it can be used to solve a number of methodically close problems (geophysical and seismic prospecting, studying ocean turbulence, etc.). In the first case, the question concerns the analogy between searches for coherent structures in turbulence and ore bodies or other geological formations [7]. In the second group of problems, it is important to study the interaction of surface and internal waves, the problem of generating volumetric turbulance in the ocean, etc. [8].

In the majority of similar situations, the frequency of data flow from the multichannel collection system to the input of the computer complex is 2-20 kHz. The duration of one experiment can be several hours, therefore, data processing must be done real-time. Processing includes filtration of the data in many variables, two-dimensional Fourier transformation, correlation analysis, etc.

To satisfy these requirements, computing power of the processing system must be on the order of 8-10 million operations a second, while in the final processing stages only floating-point computations can satisfy the requirement for a dynamic range of results.

GFVK Principles of Construction and Architecture. Evaluations show that it is currently impossible to provide the necessary capacity and processing flexibility in actual experiments, using large general-purpose computers with traditional architecture. Therefore, the problem can be solved only by creating a problem-oriented computer system.

The following principles were set down as the basis for the architecture of such a system:

modular structure with broad use of special blocks (hardware and software) to accomplish individual functions;

pipeline principle of organization, used both within individual blocks and between blocks (except the acquisition block, which is essentially a synchronous real-time system);

hardware and software equivalence, from the point of view of functions accomplished, permitting replacement of individual resources with others;

intellectualization of peripherals (storage, graphic output means, communication lines, etc.);

the capacity for simple and rapid adaptation of the system to signal and environmental variables, and also to the type of receiving device;

keeping development of non-standard equipment to a minimum; preferably use series-production domestic computer equipment and CAMAC (Computer Application to Measurement and Control) standard for connecting the computer to the subject of the experiment;

subdivided automatic testing system to provide increased reliability;

standardized connection of new devices and programs to the system.

A three-tier structure was selected as the basis of the GFVK, whereby each tier is a complete, functionally independent subsystem (Figure 2).

The lower level is designed for data acquisition, controlled by an Elektronika-60 microcomputer. It is constructed by methods adopted in the Mikro-CAMAC-lab of IAiE SO AN USSR [9, 10]. A special CAMAC module was developed that accomplishes the necessary signal transformation for preliminary analog-digital signal processing. The first level of digital data processing, included in obtaining the spatial spectrum of signals, is carried out in an Elektronika-MT/70 high-speed peripheral processor [11], connected to the common bus of the Elektronika-60 computer.

The collection subsystem (PS) is conn cted by a standard sequential communications channel to the common bus of t.ie middle level subsystem, whose purpose is to accumulate individual spatial signal spectra for the purpose of generating two-dimensional arrays and transposing them, and also for time averaging of the two-dimensional spectra obtained (then is the accumulation subsystem -- PN).

The nucleus of the GFVK is the A-12 vector processor [12], performing Fourier transformations of signals in the time field, and also secondary data processing. A standard A-12 independent input-output channel module is used to load data from the lower level into the A-12 main memory in the direct memory access mode.

Control of the vector processor, its loading and initializing, transmitting results for short and long-term storage, testing, and other A-12 system

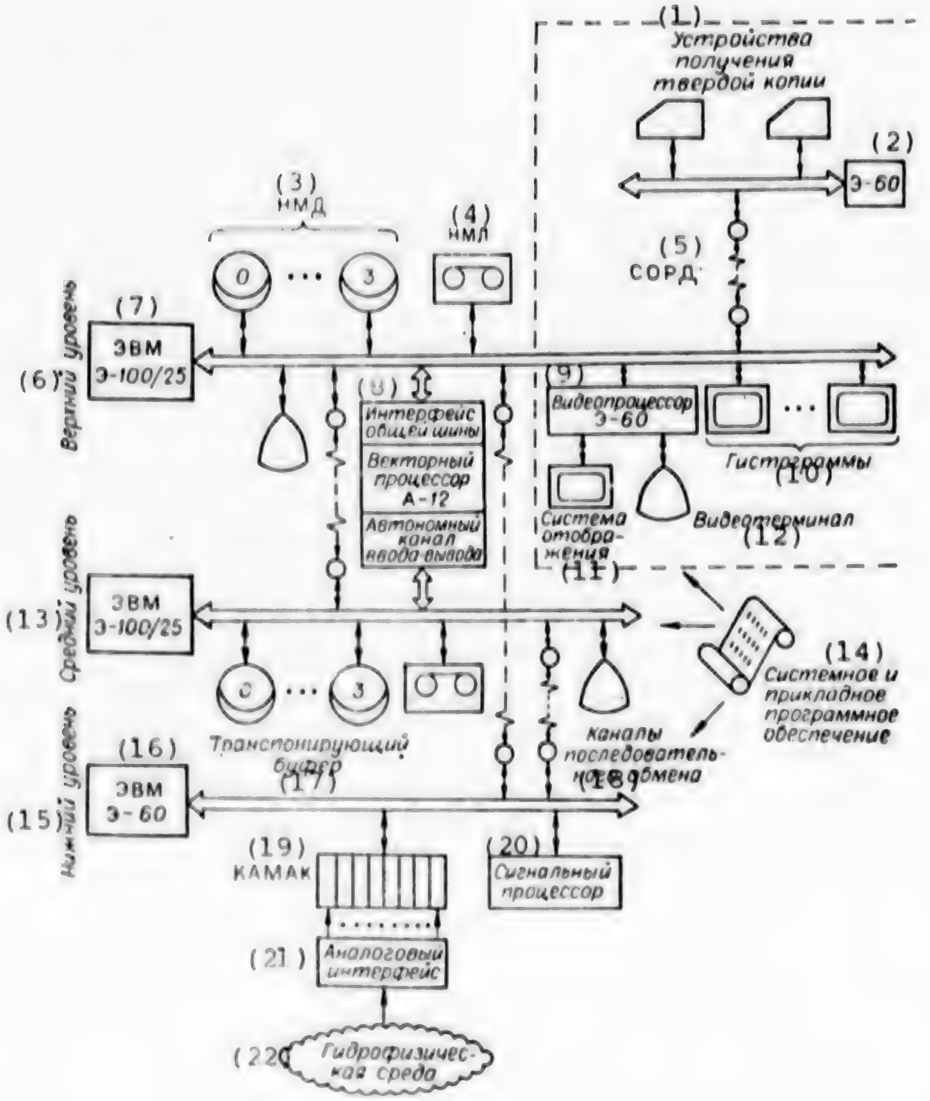


Figure 2. Structural diagram

Key:

1. Devices for obtaining hard copy
2. E-60
3. Magnetic disk storage
4. Magnetic tape storage
5. Display, recording, and documentation assembly
6. Upper level
7. E-100/25 computer
8. Common bus interface
   A-12 vector processor
   Independent input-output channel
9. E-60 video processor
10. Histograms
11. Display system
12. Video terminal
13. Middle level
14. System and application software
15. Lower level
16. E-60 computer
17. Transposition buffer
18. Sequential exchange channels
19. KAMAK (CEMAC)
20. Signal processor
21. Analog interface
22. Hydrophysical medium

support functions are carried out through the communication channel between

the A-12 and upper level subsystem. This level also is used for overall system control and synchronization of the subsystems and the flows between them (then is the control and synchronization subsystem (PUS)).

The operator's position is connected at this level, containing a number of input-output devices, constituting the display, recording, and documentation subsystem (PSORD) (alpha-numerical terminal, graphic display, character and screening device, and a dot-matrix printer for hard copy).

Organization of Data Processing the the GFVK. On the whole, the system consists of a pipeline of functional subsystems such, that the data and intermediate results of its processing flow from bottom to top -- from sensors to the display system. Each of the subsystems has its own methods for controlling individual processes, both within the individual subsystems and between them. Figure 3 shows a block diagram of all processes on-going in the system, illustrating the flow of data. The presence of such a large number of processes is due to the fact, that the majority of transmissions are done in the direct memory access mode, which is asynchronous in relation to the controlling program of the computer.

Every process, as a rule, receives data from the input buffer, modulates it, and transmits it to the output buffer [3]. The input buffer of the process will be concurrent with the output buffer for the preceeding process. Synchronization between the individual processes is done by the special flags BUFFER FULL, BUFFER EMPTY, and BUFFER READY.

The data acquisition subsystem contains the input multichannel block, switching module, analog-digital transformer, and synchronization device (timer). The subsystem is controlled by an Elektronika-60 microcomputer via the CAMAC crate-controller.

The input block is designed for analog pre-processing of input signal vectors. Depending on use, this block consists of the necessary number of individual modules, each of which has four identical channels. Each channel contains an amplifier with a programmed amplification factor and multiplier for two conjugate reference signals, generated in digital form by a separate control module. Behind each multiplier is a low-frequency filter, providing the necessary suppression of heterodyne frequencies.

The complex vector of demodulated signals obtained from the input block is transformed from analog to digital, via the assigned time intervals, and goes to the microcomputer's on-line memory. Next, this data array is sent to the Electronika-MT/70 peripheral processor.

To ensure operation of the Elektronika-MT/70 processor, a program package was developed in Macroassembler, including:

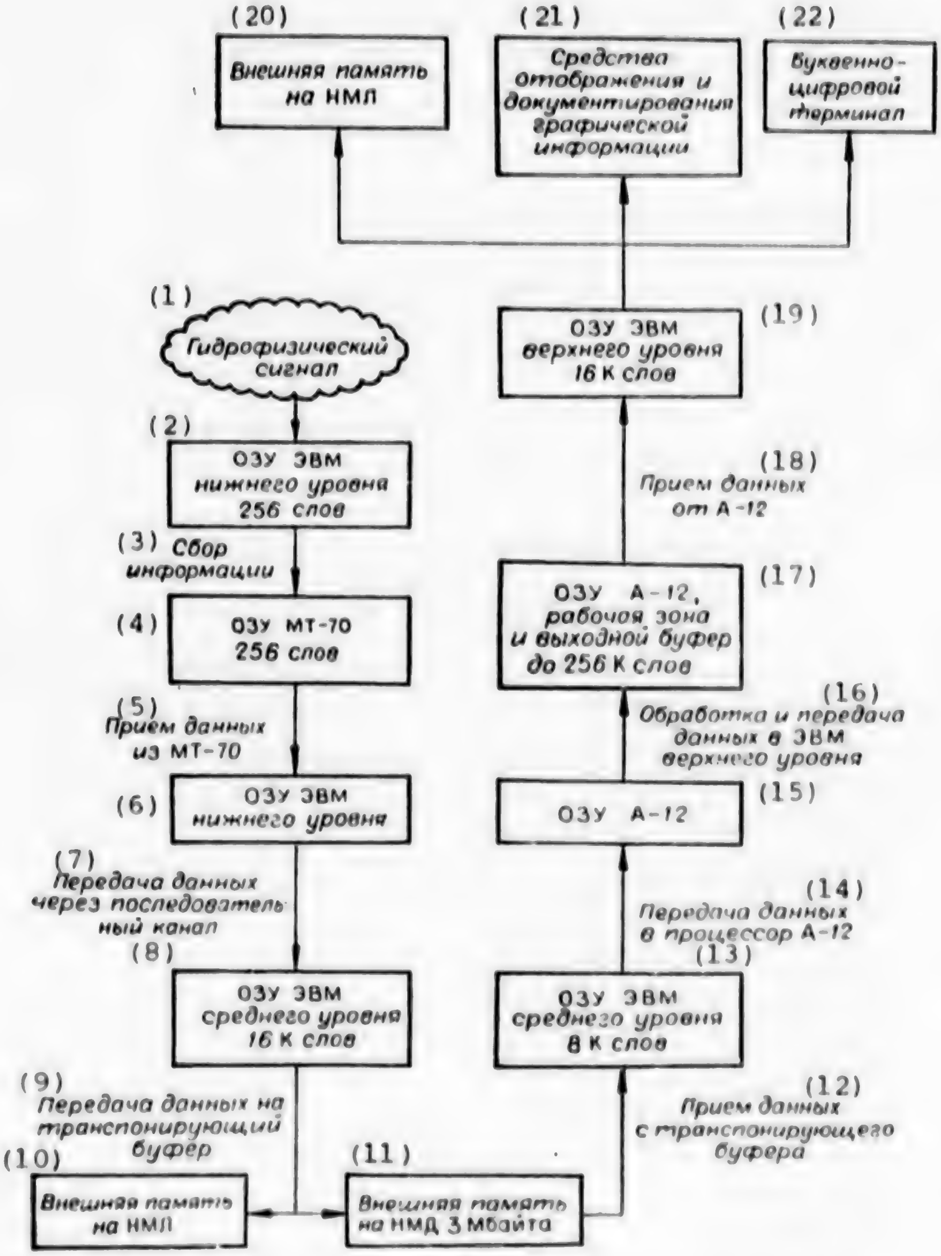

(20)
Внешняя память на НМЛ
(21)
Средства отображения и документирования графической информации
(22)
Буквенно-цифровой терминал
(1)
Гидрофизический сигнал
(19)
ОЗУ ЭВМ верхнего уровня 16 К слов
(2)
ОЗУ ЭВМ нижнего уровня 256 слов
(18)
Прием данных от А-12
(3) Сбор информации
(17)
ОЗУ А-12, рабочая зона и выходной буфер до 256 К слов
(4)
ОЗУ МТ-70 256 слов
(16)
Обработка и передача данных в ЭВМ верхнего уровня
(5)
Прием данных из МТ-70
(6)
ОЗУ ЭВМ нижнего уровня
(15)
ОЗУ А-12
(7)
Передача данных через последовательный канал
(14)
Передача данных в процессор А-12
(8)
ОЗУ ЭВМ среднего уровня 16 К слов
(13)
ОЗУ ЭВМ среднего уровня 8 К слов
(9)
Передача данных на транспонирующий буфер
(12)
Прием данных с транспонирующего буфера
(10)
Внешняя память на НМЛ
(11)
Внешняя память на НМД 3 Мбайта

Figure 3. Block diagram of processes in GFVK

Key:

1. Hydrophysical signal
2. Computer main memory, lower level, 256 words
3. Data acquisition
4. MT-70 main memory, 256 words
5. Data reception from MT-70
12. Data reception from transposition buffer
13. Computer main memory, middle level, 8 K words
14. Data transmission to A-12 processor
15. A-12 main memory

6. Computer main memory, lower level
7. Data transmission via sequential channel
8. Computer main memory, middle level, 16 K words
9. Data transmission to transposition buffer
10. External memory to magnetic tape storage
11. External memory to magnetic disk storage, 3 megabytes
16. Data processing and transmissi n to upper level computer
17. A-12 main memory, operating zone and output buffer, up to 256 K words
18. Data reception from A-12
19. Computer main memory, upper level, 16 K words
20. External memory to magnetic tape storage
21. Graphic data display and documentation equipment
22. Alpha-numeric terminal

direct access controller driver, performing exchange operations between the MT/70 data storage and the microcomputer's on-line memory;

MT/70 processor driver, necessary to transmit variables and organize the start of its microprogram;

a subroutine library, which is the interface between programs written in FORTRAN and the MT/70 drivers; this library supports the entire set of processor hardware capabilities (arithmetic and logic operations, fast Fourier transformations, statistical processing).

Organization of the sequence rests with the operating system of the RAFOS microcomputer, which makes it possible to program the MT/70 without interrupting its computations during use in the real-time mode.

Acquisition and interruption processing programs are used to manage the CAMAC hardware. The acquisition program contains, in particular, the conversion to MT/70 subroutines, using a component of the library described above for digital preprocessing of signals. The acquisition program uses a communication line driver to transmit the 32 complex spatial spectral readings (64 16-digit real numbers). Transmission is done according to the selected protocol, providing protection from desynchronization of the transmission and receive processes.

An Elektronika-100/25 computer manages the middle level of the system, and is used to accomplish preparatory operations that preceed processing. Receiving data from the lower level, transposing the array of space-time readings, and transmitting the resulting data to the A-12 main memory are among these operations. Continuous accumulation of digital readings on magnetic tape is also done here, for the purpose of maintaining information about the history of the processed signal.

To ensure functioning of this level, a program system was created, operating under the control of the real-time operating system (OS RV):

program for receiving data from the line;

program for saving partially transposed data on tape;

program for transposing data in disk storage;

program controlling the direct memory access channel to the A-12 main memory.

All of these programs are active, equal in priority with regard to the computer processor, and operate simultaneously. Flags are used for synchronization. Each flag begins its operation only having ascertained that the previous process is complete, and when its operation is ended it sets up the readiness flag, indicating the end of the process (it simultaneously drops the readiness flag of the previous process). Figure 4 shows a block diagram of the middle level.

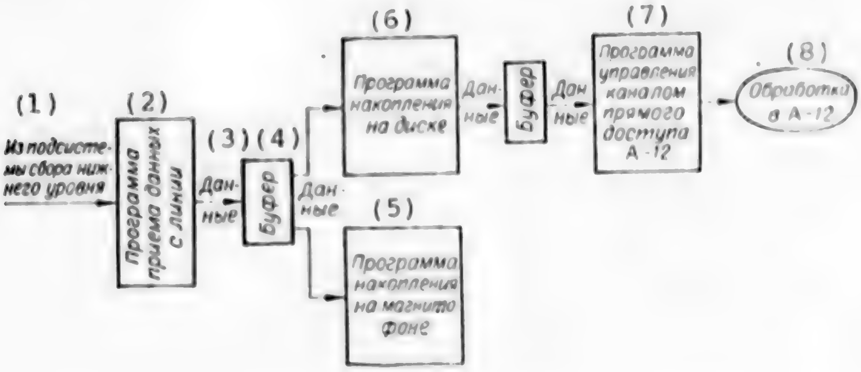

(1)
Из подсистемы сбора нижнего уровня
(2)
Программа приема данных с линии
(3)
Данные
(4)
Буфер
Данные
(6)
Программа накопления на диске
Данные
Буфер
Данные
(7)
Программа управления каналом прямого доступа А-12
(8)
Обработки в А-12
(5)
Программа накопления на магнитофоне

Figure 4. Block diagram of middle level program

Key:

1. From lower level acquisition subroutine
2. Data reception program
3. Data
4. Buffer
5. Tape storage program
6. Disk storage program
7. A-12 direct memory access control program
8. Processing in A-12

The Fourier transformation stage of time readings and subsequent accumulation of periodograms, for the purpose of obtaining spectral evaluations, is done in the A-12 vector processor. Because of the limited amount of A-12 processor main memory, the array must undergo intermediate buffering for transposition.

The transpositional buffer is based on a magnetic disk unit, in which vector readings of spatial spectra from the lower level are recorded along the radius of the flat disk, and time sequence readings to the A-12 main memory are recorded around the circumference of the disk.

Because the disk can only be accessed in blocks of 256 words, an intermediate transposition buffer of 32 X 128 complex numbers is required, established in the computer's middle level on-line memory. A data reception program, operating on a special exchange protocol, carries out data reception from the line, records acquisitions of blocks from the subroutine in 32 complex numbers in two reversing buffers (8 K volume each), one of which is always being loaded from the acquisition subsystem, and the other is writing-to-disk. The first stage of transposition is accomplished in the read-out process.

The blocks going to the cache memory on the disk are sections of time sequences that are "assembled" in a single sequence, arranged around the circumference of the disk through each of 32 write-to-disk operations. The program performing the transposition stores these partially transposed arrays sequentially in three zones, alternating in access modes such, that write-to-disk is done to one successive zone, and transposition from the two filled zones of this buffer is done to the A-12 main on-line memory.

Figure 5 represents this procedure as a process of two-stage modification of the structure of the two-dimensional array of 2 X 32 X 4 K complex numbers, whose address space is represented as two fields -- 5-digit and 12-digit. Simultaneously, the arrays coming from the lower level of the system are stored on magnetic tape.

In this way, over the course of the transposition session, the time sequences go sequentially by channels from the disk to the A-12 on-line memory.

The basic computing resources of the system is the "coupling" of the two processors of Elektronika-100/25 general purpose minicomputers with the A-12 vector processor. This union makes it possible reasonably to distribute the load to the system. The A-12 assumes the basic computing operations in carrying out the applied programs and secondary processing programs, and the upper level computer assumes management of these processes in the A-12, global control of the entire system, and ordering and displaying the results.

Computations in the A-12 are carried out in two alternating buffers (8 K complex numbers in each) in the A-12 main memory, in which arrays are recorded with an overlap of half in relationship to each other. Each buffer contains 8 overlaps per 512 words of arrays of 4 K readings.

The global control process consists of the beginning initialization of all processors, control over them as a whole, and real-time processing and transmitting reports of errors and special situations occurring in the system. Here the advantages available in the standard OS RV operating system are fully realized: management of files, operation with stored data, multiple task management, etc.

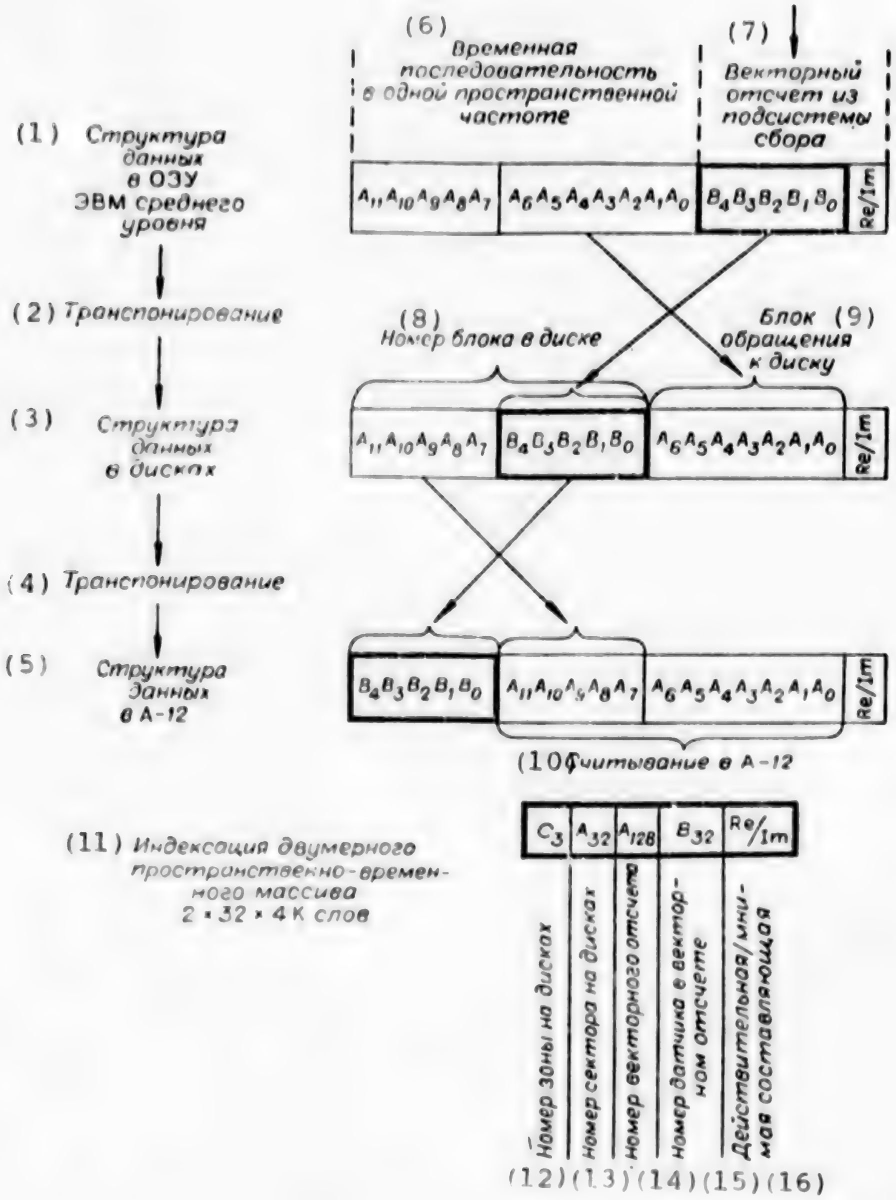

(6) Временная последовательность в одной пространственной частоте
(7) Векторный отсчет из подсистемы сбора
(1) Структура данных в ОЗУ ЭВМ среднего уровня
A11 A10 A9 A8 A7
A6 A5 A4 A3 A2 A1 A0
B4 B3 B2 B1 B0
Re/Im
(2) Транспонирование
(8) Номер блока в диске
Блок (9) обращения к диску
(3) Структура данных в дисках
A11 A10 A9 A8 A7
B4 B3 B2 B1 B0
A6 A5 A4 A3 A2 A1 A0
Re/Im
(4) Транспонирование
(5) Структура данных в А-12
B4 B3 B2 B1 B0
A11 A10 A9 A8 A7
A6 A5 A4 A3 A2 A1 A0
Re/Im
(10) Считывание в А-12
(11) Индексация двумерного пространственно-временного массива 2 × 32 × 4 К слов
C3
A32
A128
B32
Re/Im
Номер зоны на дисках
Номер сектора на дисках
Номер векторного отсчета
Номер датчика в векторном отсчете
Действительная/мнимая составляющая
(12)
(13)
(14)
(15)
(16)

Figure 5. Process of two-stage transposition in adress space of two-dimensional space-time arrays of 2 X 32 X 4 K words

Key:

1. Structure of data in main memory of middle level computer
2. Transposition
3. Structure of data on disks
4. Transposition
5. Structure of data in A-12
9. Block for access to disk
10. Read-out to A-12
11. Indexing of two-dimensional space-time array of 2 X 32 X 4 K words
12. Zone number on disks

6. Time sequence in one spatial frequency
7. Vector reading from acquisition subsystem
8. Number of block on disk
13. Sector number on disks
14. Vector read-out number
15. Sensor number in vector read-out
16. Real/Imaginary component

The display, recording, and documentation subsystem is controlled by an Elektronika-60 microcomputer, and is connected with the upper level by a sequential traffic channel. Data is displayed on a television monitor in the form of a signal level chart with technical coloration. Signal levels corresponding to each color are assigned by the operator ahead of the display process and are stored in the system in the form of tables.

The operator's position has devices for recording and documentation, as well as a control console. The operator interacts with the system in the dialogue mode: the operator can request display of portions of data arrays that interest him. Then, using the various secondary processing methods, he makes a decision about the presence of coherent structures and makes a detailed study of them.

The program subsystem for testing and detecting errors includes local and global tests. The local tests are standard program elements of all GFVK subsystems. Means of error detection, besides local software, should include the hardware (for example, parity checks in the A-12 main memory), which cause interruptions in the controlling upper level computer when there is an error, and actuation of standard programs for processing malfunctions.

The global testing system was realized by both software and hardware, and includes a space-time sequence simulator at the acquisition subsystem input. For a known input sequence, "models" of its modification are stored at each processing stage. The testing process consists of generating a full sequence and comparison, at each processing stage, of the intermediate arrays actually received with those stored in the "model" at that level.

The signal program simulator serves as a global check of paths and middle and upper level operating resources. This is a complex program, accomplished by standard system means (including the A-12), simulating reception and demodulation of a quasi-monochromatic wave packet with Gauss noise superimposed on it. The program for this complex performs mathematical modeling of signals (without real time), generation of the simulated signal from the middle level recorder and processing of the signal, dialogue with the operator assigning model variables from the console, and exercises overall management of the processes.

The simulation subsystem program makes it possible not only to test the system, but also to debug the applied software without full-scale experiments.

Conclusion. It is expedient to develop the system in the following directions:

increase the computing power of the system, mainly by intellectualization of the lower and middle layers;

expand the external storage to create archives for full-scale data and simulated signals;

replace the small computers and microcomputers with more powerful and modern models;

expand the applied software, create applied packets of programs, and make modeling of functions more complex;

improve display systems and input-output devices.

BIBLIOGRAPHY

1. "Primeneniye tsifrovoy obrabotki signalov" [Use of Digital Signal Processing], Moscow, Mir, 1981, Ch 6.

2. Monin, A.S., Yaglom, A.S. "Statisticheskaya gidromekhanika" [Statistical Hydromechanics], Moscow, Nauka, 1967, Ch 2.

3. Brandstater, A., et al. "Low-dimensional Chaos in a Hydrodynamic System", PHYS. REV. LETT., 1983, Vol 51, No 6.

4. L'vov, V.S., Predtechenskiy, A.A., Chernykh, A.I. "Bifurcation and Chaos in a Taylor Vortex System: Full-scale and Numerical Experiment", ZhTF [Journal of Technical Physics], 1981, Vol 80, No 3.

5. Eklundh, O. "A Fast Computer Method for Matrix Transposing", IEEE Trans., 1972, Vol c-72, No 80, pp 801-803.

6. Bendat, B., Pirsol, A. "Primeneniya korrelyatsionnogo i spektral'nogo analiza" [Use of Correlation and Spectral Analysis], Moscow, Mir, 1983.

7. Gurvich, I.I., Boganik, G.N. "Seysmorazvedka" [Seismic Surveying], Moscow, Nedra, 1980.

8. "Okeanologiya. Fizika okeana. T. 2. Gidrodinamika okeana" [Oceanology. Ocean Physics. Vol. 2 Hydrodynamics of the Ocean], Moscow, Nauka, 1978.

9. Nesterikhin, Yu.E., Zolotukhin, Yu.N., Livshits, Z.A. "Autometry: Results of the Decade", AVTOMETRIYA, 1984, No 4.

10. Gusev, O.Z., Zolotukhin, Yu.N., Prokhozhev, O.V., Yan, A.P. "Basic Configuration of the 'Mikro-CAMAC-Lab' System", AVTOMETRIYA, 1984, No 4.

11. Talov, I.P., et al. "'Elektronika-MT/70' High-speed Peripheral Processor", USiM, 1983, No 4.

12. Brodskiy, I.I., et al. "A-12 High-speed Peripheral Vector Processor", AVTOMETRIYA, 1984, No 4.

13. Baytser, B. "Arkhitektura vychislitel'nykh kompleksov" [Architecture of Computer Complexes], Mir, 1972, Vol 2.

12304

CSO: 1863/145

'ESTAFETA' COMPUTER-NETWORK STATION CONSOLIDATES EQUIPMENT

Moscow SOTSIALISTICHESKAYA INDUSTRIYA in Russian 13 Feb 85 p 2

[Article by I. Klimenko, correspondent (Ivanovo)]

[Text] The State Planning-and-Design Institute of Automated Control Systems has established the country's first local computer-network station, which has been given the name "Estafeta" (relay). The innovation makes it possible to bring together diverse computer equipment--universal mini- and microcomputers, displays, printers--a total of up to 125 units.

The "Estafeta" station will be used in the control of production processes and in the solution of organizational and economic problems at enterprises, while in flexible production systems such stations will link equipment with computers.

FTD/SNAP
CSO: 1863/184

UDC 681.324

MINI- AND MICROCOMPUTER LOCAL CONTROL NETWORK ARCHITECTURE AND DESIGN PRINCIPLES

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 3, July-Sept 84 (manuscript received 11 Nov 83) pp 56-58

[Article by V. A. Ivanov, candidate of technical sciences, A. A. Cherevko, engineer, and V. N. Yatsenko, engineer]

[Text] Application of the network concept to organize communication between computers allows the creation of effective local area networks (LANs) for use not only in institutes and foundations, but also as control networks in plants within various branches of the national economy [1].

The high reliability, high data transmission speeds (up to 100 Mbps), low cost and significant savings in communication lines offered by LANs permit their use as the basis for automation and control systems.

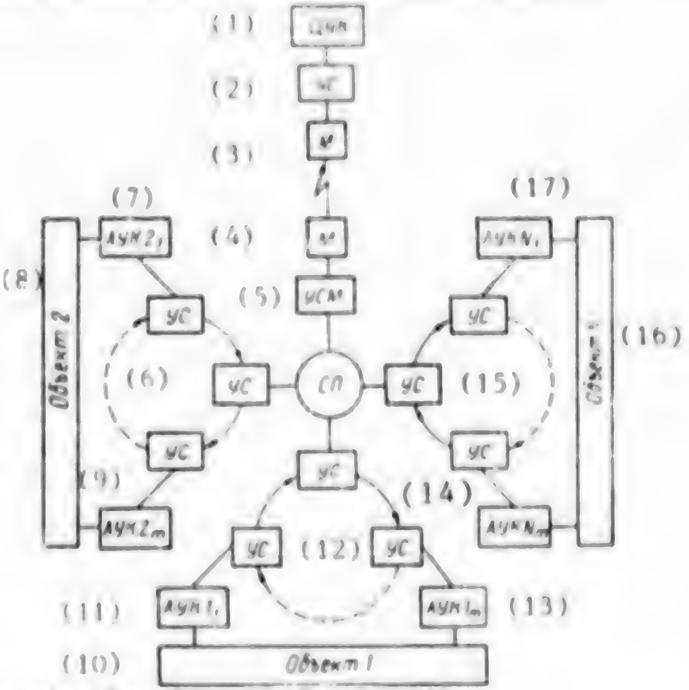
When developing an LAN architecture, one must define such items as basic logic components, interfaces and protocols, methods of control, addressing and connection, as well as diagnostics, configuration and data switching techniques. Further, one must determine the optimal relationship between hardware and software in the data transmission system. The demands to be satisfied must also be matched with such factors as the amount of delay, throughput capacity, transmitted data confidence level, reliability and cost involved.

Of the LAN configurations in current use--bus, star and ring--the latter is characterized by the greatest simplicity, lowest cost and, because of the use of repeaters, its capacity for assuring communications over the greatest distance. But the number of computers (n) connected to a common single channel (ring or bus) is limited by the value

$$n < \frac{n\tau}{\sum_{i=1}^{n} \tau_i},$$

where τ_i is the time the i-th computer uses the single channel within a monitored period τ .

A multiple ring LAN configuration has been developed at the Institute of Cybernetics of the UkSSR Academy of Sciences. Individual serial communications ring networks, each of which has $m=2^k$ interconnected subscriber control systems (where k can have a value from 0 to 8), based on the "Elektronika-60" microcomputer and the SM4, SM1420 minicomputers, are connected to a communications processor via a coupler and twisted pair wire. The communications processor acts as the controller for data interchanges between ring LANs (the number of which is $N=2^{(8-k)}$) and provides communication between local and remote terminals. Through a modem and a modem coupler, the communications processor can provide a link to a distant central control system or a regional network.



1 Central Control System
2 Coupler
3 Modem
4 Modem
5 Modem Coupler
6 Couplers
7 Subscriber Control System 2(1)
8 Object 2
9 Subscriber Control System 2(m)
10 Object 1
11 Subscriber Control System 1(1)
12 Couplers
13 Subscriber Control System 1(m)
14 Subscriber Control System N(m)
15 Couplers
16 Object N
17 Subscriber Control System N(1)
18 Communications Processor

Fig. 1. Block diagram of a multiple ring local control network

The communications processor can be concentrated, implemented on a microprogrammable processor with variable architecture or a universal minicomputer, and even distributed over N "Elektronika-60" microcomputers connected in a ring via a serial interface (Fig. 2).

Two types of hardware resources can be distinguished in the network:

1. The subscriber system, made up of subscriber and central control systems, which fulfills distributed control functions.

2) The data transmission system or switching system, carrying out all system data interchange functions, i.e. open system architecture physical-, channel- and network-layer functions [2].

Some functions of the data transmission system, the primary component of which is the coupler, can be taken up by the subscriber control system if it has sufficient reserve capacity for this task. Data transmission system design variations are possible, depending on the distribution of functions between the subscriber control system and coupler:

1. Hardware implementation of physical-layer functions in the coupler and software implementation of channel- and network-layer functions in the subscriber control system.

2. Hardware implementation of physical-layer and some channel-layer functions in the coupler and software implementation of the remaining channel-layer and all network-layer functions in the subscriber control system.

3. Hardware implementation of physical-layer and some channel-layer functions, as well as software implementation of channel-layer functions in the coupler; software implementation of network-layer functions in the subscriber control system.

4. Hardware implementation of physical- and channel-layer functions, as well as software implementation of network-layer functions in the coupler.

Since the task of assuring data interchange within the network is carried out by a combination of hardware and software resources, the selection of an optimal relationship between these elements is one of the most important problems in data transmission system design. The suitability of any data transmission system design depends on system throughput capacity and reliability demands, cost, and the number of communication lines, as well as subscriber control system and coupler traffic levels. Suitability can be defined by the efficiency criterion Ke, representing throughput capacity relative to cost unit, including reliability and communications line loading factors:

$$Ke = \frac{\gamma T(1-R)}{c1 + c2 + \alpha c3 + \beta c4},$$

where T is the data transmission system's throughput capacity, R is the bit distortion confidence factor during data interchanges, c1 is the cost of data transmission system software resources in the couplers and subscriber control systems and c2, c3 and c4 represent the cost of coupler, subscriber control system and subscriber system memory hardware. α and β, respectively, are the subscriber machine time and memory usage coefficients for the execution of data transmission system functions. $\gamma = \frac{tp}{t}$ is the communication line traffic coefficient and tp is the time taken up by data transmissions during an observed time interval t.

Since current mass-produced couplers for mini- and microcomputers only carry out physical functions and are not oriented toward network architecture, they can be used only as a basis for LANs with a star configuration, using the first data transmission system variant which has the lowest speed.

The third data transmission system design variant is characterized by the use in couplers of current general-purpose microcomputers and microprocessors which have the drawback of limiting network data transmission speeds to a maximum of 50 Kbps and require software implementation of a communications protocol in the microcomputer language. This was the reason behind the development of specialized large-scale integrated circuits (LSICs) supporting standard protocols which carry out physical- and channel-layer functions. Thus, the fourth data transmission system variant came about with the use of specialized LSICs in couplers and microcomputers. Among these LSICs are the Intel 8273 (8274 and 82586), Western Digital 1933 (2511), Zilog 510, Signetics 2652, Motorola 6854 and others.

1	Microcomputer 3
2	Couplers
3	Microcomputer N
4	Microcomputer 1
5	Microcomputer 2
6	Communications Processor

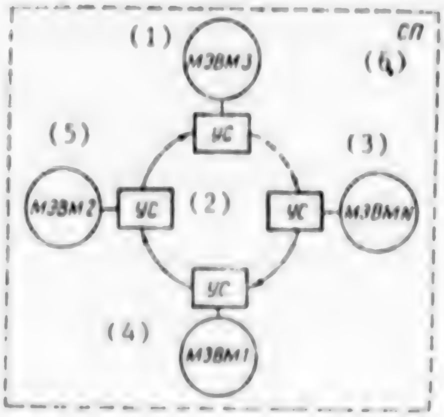


Fig. 2. Block diagram of a distributed communications processor

By analyzing the data transmission system function algorithm and measuring the speed of communication line and computer operations during coupler development, a conclusion was made with regard to the economic suitability of the second data transmission system design variant, with hardware implementation in the coupler of those functions whose software implementation would involve considerable amounts of time. The coupler provides asynchronous communication at rates of up to 9.6 Kbps and synchronous communication at up to 56 Kbps with a 1 km separation between connected subscribers.

The coupler consists of two parts: the network section which performs open-system architecture physical-, channel- and network-layer functions and the subscriber section which acts as the interface between the subscriber control system and the network section.

The network section, or the local network interface, performs several basic X.25 communications protocol functions. It assures bit synchronization by detecting the clock frequency in the receiver; equalizes the network transmission rate to the subscriber control system speed by means of an 8-bit buffer, matches interface logic level signals to the transmitting unit's signals through transmit and receive amplifiers; provides cycle synchronization by generating and detecting a 01111110-type flag to designate the start-of-block, converts parallel code to serial and vice versa, supports communications transparency through "bit stuffing"; monitors data through parity and cyclical code checking; detects control sequences, controls transmission by recognizing the control marker, converting it to a flag and subsequently issuing its message; controls reception by detecting the address field, recognizing its address and transmitting any data received in this case to the subscriber section; relays received data further along the ring.

Two types of couplers have been developed. They have identical network sections and are distinguished by their subscriber sections. The coupler can work in the duplex or half-duplex modes through a two-point connection to a ring, star or multiple ring network.

Sixty small- and medium-scale integrated circuits, mounted on a single board, are used in each type of coupler.

The self-synchronizing ternary code used for data transmission in the LAN allows a significant reduction in coupler hardware outlays and simplifies the task of bit synchronization in the LAN.

The coupler is fitted with a relay to prevent ring interruption in case of a subscriber control system power outage.

Packet switching is incorporated in the network. A modified bit-oriented HDLC protocol is used for data channel control.

Data are transmitted in blocks with a maximum size of 256 bytes. The block format is F, AP, AI, U, D, I, K, M; where F is the flag, AP is the physical address of the receiver, AI is the physical address of the source, U is the control element, D is the block length in bytes, I is the data, K is the check element and M is a marker.

The data field can contain any sequence of bits since the coupler supports data transparency. The two-byte check field holds the remainder produced through multiplication by X^{16} and subsequent division (modulo 2) by an original polynomial $X^{16}+X^{12}+X^{5}+1$ of the contents of the block's AP, AI, Y, D and I fields.

The low-order bits of the receiver address field specify a subscriber control system address and the high-order bits of this field designate the ring network address.

The "token-passing" ring method was chosen from among the various data transmission medium access methods proposed by IEEE Standard 802 for local area networks. With this method a marker is used to control data transmission. Thus, a subscriber can transmit data only after receiving a "token"--or control marker.

The procedures required to establish, maintain and terminate connections to the ring network are carried out by the coupler, together with the subscriber control system and communications processor. Prior to the establishment of a connection, a marker introduced by the communications processor circulates within each ring network. When a data block is ready for transmission, the coupler, relaying received bits further along the ring, awaits the arrival of a 11111110 marker. Upon receiving the marker the coupler changes it to a flag by inverting its final bit and sends the message to the ring network. On receiving a flag each coupler detects and analyzes the receiver address field. When it recognizes its own address, the coupler receives the message for its subscriber control system and relays it to the next coupler.

After receiving confirmation of receipt, the transmitting subscriber control system issues a marker used as a flag to break the connection to the network and to permit data transmission by another subscriber control system. If the message is intended for another ring network, it is latched in the communications processor and sent to the addressee after the destination network is freed, i.e. after receipt of a marker from that network.

Two types of addressing are used in the network: fixed and broadcast. With broadcast addressing, messages are sent to a group of addressees rather than a single addressee. Both types of addressing are implemented in the coupler by means of an EPROM.

Possible marker loss in a network is established by means of a marker appearance timer whose duration depends on block length, transmission rate and number of subscribers in the network.

Network operational experience has confirmed its high reliability and efficiency. The LAN examined is currently being introduced in one of this country's industrial enterprises.

1. Yakubaytis, E. A. et al.,"Arkhitektura lokal'nykh vychislitel'nykh setey" [Local Area Network Architecture]. AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA, 1983, No 2, pp 3-20.

2. Yakubaytis, E. A., "Arkhitektura vychislitel'nykh setey" [Computer Network Architecture], Moscow: Statistika, 1980, 279 pp.

12746
CSO: 1863/66

UDC 681.3:578.088.78

HARDWARE AND SOFTWARE OF STANDARD MEASURING CHANNEL FOR MICROELECTRONICS RESEARCH

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 9 Aug 84) pp 44-52

ASTAF'YEV, S. V., TRET'YAKOV, V. P., SHTARK, M. B. and YANOVSKIY, G. Ya., Novosibirsk

[Abstract] An automatic data gathering and preprocessing system with an "Elektronika-60" microcomputer and CAMAC modules is described which features flexibility and adaptability necessary for experimental research in microelectronics. It is designed for handling voltage signals of microvolt to millivolt magnitude at frequencies up to 20 kHz through two of its channels and current signals of 1-5000 pA magnitude in the same frequency range through a third channel. It records up to 16 signals and gives up to 16x1024 readings per measurement cycle, the signal discretization period ranging from 10 μs to 32 ms. In addition to the CAMAC crate, the hardware includes storage on flexible magnetic disks, a terminal, a printer, and a graphic display with memory tube. The software includes an information retrieval system which, in addition to rigid programs for executing the algorithms of experiments, contains an assortment of auxiliary functions such as hookup for the dialog mode of operation, system debugging, on-line structure and process modification, hardware inspection, and conversion from one mode of data display to another. Instructions include HULLO and FORMST, ACTION and CHANNEL, GLAD and PLOTER for preparatory operations such as CLEAR and INITIATE, computation processes such as TRANSLATE and STIMULATE, followed by data finishing operations such as smoothing and displaying respectively. Included are also operations CALIBV and CALIBI of voltage calibration and current calibration as well as calibration BALANC of the microelectrode impedance. The dialog language reflects the methodology of experiments and their evaluation, it can be modified and supplemented without additional programming or editing of individual modules. Figures 4; references 8: 5 Russian, 3 Western (1 in Russian translation). [74-2415]

UDC 629.7.058.74:681.3.06

COMPUTER-AIDED SYNTHESIS OF ENVIRONMENT PICTURES

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 16 Jan 84) pp 67-76

KOVALEV, A. M. and TALNYKIN, E. A., Novosibirsk

[Abstract] Images of visual objects are synthesized by either physical or mathematical simulation. While physical simulators require mechanical carriers of information and are limited in scope, mathematical simulators feature both versatility and flexibility. The main criterion is retention of objective and psychophysiological similarity. The problem has been formulated in terms of an observer-medium-objects system. A model of its solution has been constructed which involves several successive operations: mapping all objects into a single system of coordinates, extraction of objects within the field of vision, projection of these objects onto the plane of viewing, removal of invisible surfaces of hidden objects, computation of surface colors, formation of image signals and their transmission to the imaging device such as a television monitor. The necessary computer graphics are based on approximation of surfaces and hierarchical structurization of object models, parallel computations and proper layout of the computation pipeline, and efficient algorithms implementable on special-purpose calculators in the pipeline. The pipeline consists accordingly of four successive stages. The input stage is a scene processor, which transmits global data to the geometrical processor. The latter generates local data and transmits them to the video processor for projection onto the viewing plane. The last stage shapes video signals from elements of the projected image. Prototypes of two pipelines for synthesis of pictures have been developed, built, and tested at the Siberian Department of the USSR Academy of Sciences. The simpler first one operates with an "Elektronika-100/25" computer as special processor, using a disk storage for global data and the memory in the "Elektronika MT/70M" peripheral special processor for storage of local data. The second one is designed for more intricate objects and images, for processing over several video channels with adequate distribution of available capability. It has a built-in protection against data overflow. It is capable of imitating smoke, fog, and clouds in accordance with ICAO standard weather thresholds. It also includes digital filtration for better image quality and a monitoring-diagnosing subsystem. The software for real-time data processing for viewing fast and simultaneously occurring events requires nonstandard operating systems matched to television line and frame frequencies. These are the major problems involved in development of picture synthesizing systems, to which must be added cost considerations. Figures 4; references 10: 4 Russian, 6 Western (1 in Russian translation).
[74-2415]

UDC 681.3.06

PRINTED-CIRCUIT DESIGN WITH AID OF 'ELEKTRONIKA'-FAMILY COMPUTERS

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (mansucript received 23 Jan 84) pp 89-93

TALNYKIN, E. A., Novosibirsk

[Abstract] A system for designing printed circuits with the aid of "Elektronika" and SM-4 computers is being developed at the Institute of Automatic Control and Electrometry, Siberian Department of the USSR Academy of Sciences. Its principal components are a design slide containing a set of the simplest "structural" elements, a system of nondiscretized Cartesian coordinates locating the printed-circuit board in the upper positive quadrant, a rectangular grid with appropriate X and Y discretization steps superposable on the printed-circuit board, a technological slide containing a set of tools for adaptation of the design to manufacturing capability, and postprocessors containing software modules for numerical program control of machine tools which produce printed circuits. The boards can be single-layer or multilayer (2-15), each circuit element definitively located in the coordinate system constituting a "printed" element and a conductor being the trace which an element of the design slide leaves on the board while it is moved from one position to another. The language of slide description includes words defining various forms of mask (round, square, rectangular) and drills for holes as well as their locations and movements. Design and nameplate information is generated and visibly imprinted on the outer board layers, for identification and treatment purposes. The author acknowledges stimulating discussions with V. A. Meleshikhin, D. G. Frizen and Ye. G. Yurashanskiy. Figures 3; references: 3 Russian.
[74-2415]

UDC 621.3.049.771

HYBRID CIRCUITAL AND FUNCTIONAL-LOGIC SIMULATION OF ANALOG-DIGITAL DEVICES

Kiev ELEKTRONNOYE MODELIROVANIYE in Russian No 5, Sep-Oct 84 (manuscript received 3 Feb 83, after revision 20 Apr 83) pp 35-39

ARKHANGEL'SKIY, ALEKSEY YAKOVLEVICH, candidate of technical sciences, docent, and MELIKYAN, VAZGEN SHAVARSHOVICH, graduate student, Moscow Institute of Engineering Physics

[Abstract] Hybrid circuital and functional-logic simulation has been proposed for modeling of LSI analog-digital devices, pure circuital simulation based on solving systems of differential equations not being feasible here for design purposes. Hybrid simulation involves concurrent solution of differential equations describing the analog component of a device and solution of logic equations for its digital component. The three principal problems in such a simulation are discontinuity of functions with resulting instability or slowdown of the computation process, restrictions imposed by delays in the logic

model on the size of time step in the computation grid, and poor accuracy of models at the transition from logic component to analog component. A modification of this method is proposed here which largely overcomes all three difficulties. The analog-digital device is partitioned into an electric circuit and a logic module, the former including not only the analog component but also fragments of the digital component requiring precise signal waveform determination and the latter containing that part of the digital component which requires precise computation of time characteristics with less emphasis on the signal waveform. The method is demonstrated on the specific case of an implicit scheme for numerical integration of differential equations with automatic step selection and variation, these equations being formulated for node potentials with necessary expansions of the coordinate basis. The efficiency of this method is evaluated for two LSI analog-to-digital converters, an 8-digit one containing approximately 200 components and a 12-digit one containing over 1000 components. Simulation on a YeS-1033 computer with various sequences of input signals required 3-10 min machine time. Separate simulation of the analog component and the digital component would require over 10 h and for the former and tens of hours for the latter. The method has also been tested on several other LSI circuits containing up to 1300 transistors. Figures 2; references 4: 2 Russian, 2 Western.
[106-2415]

UDC 772.61:771.534:531:429:621.391:681.515.8

METHODS AND MEANS OF ON-LINE DIGITAL IMAGE PROCESSING

Novosibirsk AVTOMETRIYA in Russian No 4, Jul-Aug 84 (manuscript received 21 Mar 84) pp 97-102

KIRICHUK, V. S., KOSYKH, V. P., NESTERIKHIN, Yu. Ye. and YAKOVENKO, N. S., Novosibirsk

[Abstract] Problems in producing components of "wide-band" on-line image processing systems, specifically digital processing systems, arise in both hardware and software design. The principal requirement is flexible multifunctionality, with an adequate array of input devices covering the entire range of possible data forms. Another important requirement is availability of adequate displays interacting with large video data arrays, especially when heuristic algorithms are to be used for data processing. The main difficulty lies in the inadequacy of conventional structures based on otherwise highly efficient general-purpose computers. Special-purpose arithmetic and logic processors are need for performing such classes of operations on continuous amplitude corrections involving several images, local transformations on individual images where the transform depends on the optical density of the original at a given point as well as in its vicinity, morphological transformations of images, integral transformations, and geometrical transformations. The processing system configuration based on "Elektronika" family computers with "common bus" of the older models is most promising for this application, having a potential capacity of 10^8 operation/s and 32-64 Mbyte/s. A typical problem which the system should be able to solve is

scanning a series of images for distinctions. This requires first plotting all images in a single system of coordinates, after successively rough and fine estimation of identical elements as reference points, then determining the geometrical transformation of images which will ensure their conformance. This is followed by amplitude correction of the images for compensation of differences in brightness recording. The least standard past part of the processing algorithm here in search of significant distinctions with their subsequent classification. The present trend in development of image processing systems is design of maximally utilizable special-purpose high-efficiency processors and structures, with attendant design of problem-oriented software. References 20: 15 Russian, 5 Western (1 in Russian translation).
[74-2415]

UDC 681.324

OPTIMIZING CONFIGURATION OF MACHINES, PROGRAMS AND FILES IN COMPUTER NETWORK

Riga AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA in Russian No 5, Sep-Oct 84 (manuscript received 6 Apr 84) pp 14-20

YANBYKH, G. F.

[Abstract] A method of optimizing the configuration of machines, programs, and files in a computer network is proposed, on the assumption that all channels of communication form a completely interconnected structure with subscribers' computer terminals located at its nodal points. The problem of optimizing the locations of host machines, programs, and files for a special-purpose computer network is solved on the basis of a mathematical model containing three sets of variables associated respectively with computer terminals (X_{is}), programs (Y_{js}), files (Z_{ks}), each subject to a constraint: each computer terminal is connected to only one host machine, each program is placed with at least one host machine, and each file is placed with only one host machine. In addition, all variables are Boolean and the mean response time of the computer network must not exceed the maximum permissible length. The proposed algorithm is a heuristic one based on the principle of coordinate-by-coordinate optimization. It consists of four steps: location of subscribers' programs and files--initial location of systemwise programs--initial location of systemwide files--rounding up systemwide programs and files for refinement of the optimization. After the optimization problem has been solved, one can determine the necessary output capacity of each host machine on a scale of discrete values and the transmission capacity of each communication channel. The results are demonstrated on two specific examples. In the first example each of three control moduleshas access to only one of three programs and each of the latter has access to only one of three files, all programs being identical with respect to functions and size-time characteristics. In the second example each of three subscriber programs has access to any of three different systemwide programs and to any of four different systemwide files, while each of three systemwide programs has access to any of three systemwide files but not to any of three subscriber files. Figures 4; references 11: 7 Russian, 4 Western (1 in Russian translation).
[104-2415]

UDC 681.324

ADAPTIVE CONTROL OF MEMORY IN YeS-7920 INTERFACE CONVERTER

Riga AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA in Russian No 5, Sep-Oct 84 (manuscript received 5 Mar 84) pp 21-23

KOROBKOV, B. P. and RATSIN, Yu. V.

[Abstract] An adaptive algorithm has been constructed for static allocation of memory in a YeS-7920 interface converter. Such a converter is a special-purpose computing device on the basis of an SM-1800 microprocessor and a network module, data exchange between them being effected according to the LAP/B protocol with maximum capacity of 131 bytes. Data exchange with subscribers proceeds according to the BSC protocol with maximum capacity of 2 kbytes. The direct-access memory of the SM-1800 microcomputer is designed to meet these requirements, yielding as many as 100 caches of 131 bytes each. The algorithm reserves memory capacity for data arrays from a YeS-7920 computer and for data batches from the network module, then clears the memory after confirmation that data packets have been successfully transmitted through the computer network. The operations involved in sending and receiving data are analogous to operations RQWAIT and RQSEND in interproblem interaction in a real-time modular on-line system. The algorithm is designed to prevent blockage in the memory allocation layout, which could happen when the registers of free caches and the reserve register for receiving data from the network module have been exhausted before the end of a message cycle. References 2: 1 Russian, 1 Western.
[104-2415]

UDC 621.394

PREDICTION PROTOCOLS FOR RANDOM MULTIPLE ACCESS IN COMPUTER NETWORKS WITH PACKET TRANSMITTING RADIO COMMUNICATION CHANNELS

Riga AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA in Russian No 5, Sep-Oct 84 (manuscript received 5 Mar 84) pp 36-41

VOYTER, A. P. and OFENGENDEN, R. G.

[Abstract] The problem of nonconflicting transmission of data packets over radio communication channels in multiple-access computer networks is now tackled by protocols, with or without carrier check, which predict conflicts and delay transmission to avoid them. Two new protocols are proposed here, both synchronous ones for the channel operation period divided into cycles of duration equal to the maximum signal propagation time between subscriber terminals. According to the first protocol, which predicts without urgency (p= 0), a subsctiber delays an attempt of transmission and repeats it after a time interval of random length. According to the second protocol, which predicts with urgency (p= 1), a subscriber checks the state of the channel in each following cycle and waits for channel clearance to attempt transmission.

The mean transmission rate, defined as the ratio of mean time of nonconflicting packet transmission within a restoration cycle to mean duration of that cycle, is determined from operation timing diagrams. Analysis of these diagrams and performance calculations are based on the theory of channel restoration and the model of infinitely many subscriber terminals generating data packets of infinitesimal intensity each. A comparative evaluation of these two protocols relative to conventional ones in terms of throughput capacity as well as mean transmission rate and mean transmission delay reveals their advantages over nonsynchronous p= 0 and p= 1 protocols respectively, even though they are not optimal over the entire range of variation of system parameters (0 p 1). Figures 5; tables 1; references 6: 3 Russian, 3 Western (1 in Russian translation).
[104-2415]

THEORY OF COMPUTATIONS

UDC 681.325

INPUT OF CONSTANTS TO LARGE-SCALE-INTEGRATION CHIPS OF LOGARITHMIC CALCULATORS

Leningrad IZVESTIYA VYSSHIKH UCHEBNYKH ZAVEDENIY: PRIBOROSTROYENIYE in Russian No 10, Oct 84 (manuscript received 10 Nov 83) pp 31-36

BAYKOV, V. D. and KRYS', A. I., Leningrad Institute of Electrical Engineering imeni V. I. Ul'yanov (Lenin)

[Abstract] Two problems in constructing a calculator with LSI bipolar noncommutated logic arrays for taking logarithms and finding antilogarithms to various bases are selecting a universal calculation procedure and organizing a parallel input of initial conditions and of constants. A typical example is parallel input of constants $C_i = \log_b(1\underline{+}2^{-i})$, which is done in two stages in accordance with recurrence relations so that initial conditions $X_0= X$, $Y_0= 1- X$, $\theta_0= 0$ ($X= [0.1,1]$, 0_i- current value of the function on i-th step) yield the result $f(X)= 0_{n-1}= -\log_b X$. An analogous procedure is followed for finding antilogarithms. Universality of the calculator requires input of interchangeable constant C_i during setup for calculation of a function to a definite base b, sequential input being time inefficient. Preferable parallel input of n m-digit constants is effected in a binary code and requires $N= mXn \simeq m^2$ external contacts to the array. With not more than 56 contacts in a noncommutated logic array, m is the positive root of the equation $N= (2m+ 3) + m^2 = 56$ or $m^2 + 2m- 53= 0$ rounded to the nearest smaller integer $-1+\sqrt{54} \gtrsim 6$. The number of memory cells and thus the number of external contacts, also the access time to the read-only memory, can be reduced by application of the digit-after-digit iteration method with a smaller number P of constants. Depending on the number n of iterations, $P_{min} = 0.5n - 0.5m$ and then $N_{min}= (2m+ 3)+ 0.5m^2$. Furthermore, 12-digit constants $C_1= \log_2(1+2^{-i})$, $C_2= \log_e(i+2^{-i})$, $C_3= \log_{10}(1+2^{-i})$ can be reduced to 10-digit ones, upon elimination of coincidences in their sixth and eighth digits, so that in this case $N_{min}'(2m+ 3)+ 0.3m^2$. The same principle applies to constants $C_4= \tan^{-1}2^{-i}$, $C_5= \tanh^{-1}2^{-i}$ for calculating elementary functions sinX, cosX, tanX, cotX, tanX, $\tanh^{-1}X$ by this method. The digit-after-digit iteration method of calculator design yields the maximum possible uniformity of cells, which is an important technological advantage. Afticle was recommended by Department of Computer Engineering. Figures 2; tables 3; references: 3 Russian.
[90-2415]

UDC 681.327

CHARACTERISTICS OF MOORE MACHINE IMPLEMENTED WITH ARRAYS

Leningrad IZVESTIYA VYSSHIKH UCHEBNYKH ZAVEDENIY: PRIBOROSTROYENIYE in Russian No 10, Oct 84 (manuscript received 20 Mar 84) pp 43-46

BARKALOV, A. A. and DZHALISHVILI, Z. O., Leningrad Institute of Precision Mechanics and Optics

[Abstract] A microprogram Moore machine for given sets of states, input variables, and output variable is built according to a structural table with coded initial state, next state, and output columns. Given are also the memory excitation functions and microinstructions contained in the initial state. The logic scheme of this machine is described by the corresponding system of Boolean functions. The trivial structure of such a Moore machine consists of array M_X converting input variables to F-terms, array $M_{\emptyset}$ converting F-terms to memory excitation functions, memory Pr from which state code conjunctions are fed back to array M_X and fed forward to array M_A forming state conjunctions, and array M_Y converting state conjunctions to output variables. The complexity of such an implementation depends on that of the arrys and usually has a large redundancy. It is possible to minimize the area of array M_X and thus the redundancy of this Moore machine implementation by adding a Boolean converter of state codes directly between array M_X and array M_Y in parallel with the $M_{\emptyset}$-Pr-M_A sequence. The structural table is transformed accordingly into a γ-table, the transformation being based on states of the equivalent Mealy machine. The structure of the Moore machine ceases to be trivial and the complexity of its implementation is reduced,on the average, by 30%. Article was recommended by Department of Computer Engineering. Figures 2; references: 2 Russian.
[90-2415]

UDC 681.51

ESTIMATING COMPLEXITY OF ORGANIZATIONAL CONTROL STRUCTURE

Dushanbe DOKLADY AKADEMII NAUK TADZHIKSKOY SSR in Russian Vol 27, No 2, Feb 84 (manuscript received 23 Nov 83) pp 69-72

LIMONOVA, T. I., Council for Study of Productive Resources, USSR State Planning Committee, and MUSAYEVA, R. G., Tadzhik Polytechnic Institute

[Abstract] In planning and design of an organizational control structure for a given object there arises the problem of complexity, the diversity of possible variant to be chosen from increasing in an avalanche manner as the number of control levels is increased. The ultimate choice of control structure will depend on the particular purposes of control and its effectiveness will depend on the system parameters and on interaction between system elements as well as on external influencing factors. Complexity or its inverse,

simplicity, is one of the four principal system characteristics, the other three being reliability, speed, and stability. Simplicity, i.e., least complexity requires minimizing the number of couplings necessary for performing the given task, while reliability is most readily increased by addition of redundant couplings. In the planning and design stage, therefore, it becomes necessary to evaluate the complexity of structural control variants. Here a method of such an evaluation is proposed. It evolves from the canonical equation for the feedback operator $\tilde{k}$, which quantitatively determines and characterizes a control structure in terms of control levels and controlling elements as well as couplings and interactions. The corresponding complexity criterion $C = \prod_{k_i = E,\ i = \{1,2,\ldots,n\}} \tilde{k}$ is derived on this basis and then reduced from algebraic to arithmetic form for numerical evaluation. Its calculation according to this procedure is illustrated on a maximally complete 3-level control structure for a single object. It is found to be proportional to the length of system functioning time. A hierarchical structure with fewer control levels is recommended where efficiency is an over-riding requirement. Article was presented by Academician Z. D. Usmanov, TaSSR Academy of Sciences, on 6 May 1983. Figures 2; references: 3 Russian.
[95-2415]

UDC 657.6.012.16:519.233.8.6:519.248

FORMATION OF TARGET FUNCTION AND EVALUATION OF EXPERT ACTIVITY IN ITERATIVE DATA GATHERING AND PROCESSING

Moscow DOKLADY AKADEMII NAUK SSSR in Russian Vol 278, No 3, Sep 84 (manuscript received 5 Dec 83) pp 568-570

PETROV, V. V., corresponding member, USSR Academy of Sciences, Moscow Institute of Aviation imeni Sergo Ordzhonikidze, and KHATIASHVILI, V. G., Scientific Research Institute of Experimental and Clinical Surgery imeni K. D. Eristavi, Tbilisi

[Abstract] In the absence of a theory on formation of target functions for iterative data gathering and processing, this must be done on an experimental basis. Here one such experiment is described, with a set of multidimensional objects or measurements corresponding to a given set of indicators (criteria, tests, symptoms). Elements of that set are assumed to appear in different scales and p relations $R($ $A \times A$ are assumed to have been established by p experts on the basis of their "black box" knowledge. The unknown target function is assumed to be smooth, any two objects are assumed to admit comparison, and a linear or quasi-linear order corresponds to the relation $R = P \cup I$. To all objects in series P are assigned "ball park" estimates and a second kind ($j = 2$) of expert evaluation is performed, with ranking and modeling of heuristic objects ($b \in B$ ($C = A \cup B$) satisfying at least one of three applicable constraints, followed by a third kind ($j = 3$) expert evaluation $A \bigcup_{\substack{j=1\\ j \neq i}}^{P} B_j = c_i$". The "errors of each y_i reading are assumed to have all

almost the same dispersion over the entire range of measurement. Then a system of individual r_{ij} estimates of second kind and third kind are obtained from each expert. Now the competence of each expert must be evaluated and standard statistical methods, including regression analysis, applied to the data processing. Correct application of these methods implies a normal distribution of experts' estimates and, therefore, agreement between experts serves as a test of two simultaneous hypotheses: linear relation interrelation between data reported by experts. A test result within the given confidence interval indicates existence of a single consistent target function. A test result not within that confidence interval indicates existence of several target functions which, for estimating purposes, must be added up with weight factors normalized to the experts' competence. References: 8 Russian.
[97-2415]

UDC 51:621.391

CONSTRUCTION OF COMPLETE SYSTEM OF EXAMPLES FOR ONE CLASS OF PROGRAMS

Moscow DOKLADY AKADEMII NAUK SSSR in Russian Vol 278, No 3, Sep 84
(manuscript received 5 Dec 83) pp 564-568

AUZIN'SH, A. I., Latvian State University imeni P. Stuchki

[Abstract] The problem of constructing a complete system of examples for a new class of programs, recurrent ones, is considered and solvability of this problem is proved on the basis of three pertinent theorems. The concept of recurrent automaton as generalization of a plain finite automaton is formulated in terms of the pentad $\{Q, \Sigma, q_0, F, \delta\}$ (Q- set of states, Σ- set of symbols on data carrier tape including empty symbol $\lambda \in \Sigma$, q_0 - initial state, F- set of final states, δ- transition function mapping set Q X Σ in set Q X Σ). Program is defined as a schematic graph constructed from seven instructions (START, $A \to x \pm$, $x \to A$, NEXT A(A), $x \to y$, $x < y$, STOP). START is repeatable, the head does not have to shift after each recording or reading event, and all tapes do not have to be either incoming (reading) or outgoing (recording) ones in recurrent programming language. Example is defined as a composition presented to each tape according to a particular data array, a complete system of examples being one where any outcome of any instruction realizable in any example whatever is realizable in any example of this system. The first theorem pertains to solvability of the emptiness problem and insolvability of the equivalence problem for a recurrent automaton. The second theorem pertains to existence of an algorithm of constructing a finite complete system of examples for any program in recurrent programming language. The third theorem pertains to existence of an algorithm of constructing a complete system of examples for any program in recurrent programming language by the direct access method. The author thanks Ya. M. Barzdin' for attentiveness. Article was presented by Academician A. N. Kolmogorov on 15 November 1983. Figures 3; references 5: 4 Russian, 1 Western.
[97-2415]

SYNOPSES OF ARTICLES IN MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 3, July-Sept 84 pp 63-64

UDC 658.012.011.56:(621,771.02:621.78).001.57

IMITATIVE MODELING OF METAL HEATING IN CONTINUOUS FURNACES

[Synopsis of article by L. L. Pryadkin and V. N. Tkachenko, candidates of technical sciences, pp 1-4]

[Text] An imitative model was created to improve an automated control system for technological processes used in heating metal prior to rolling. The model was used to investigate the probability characteristics of temperature fields and to evaluate control method effectiveness according to a heating precision criterion. Two illustrations, two references.

UDC 631.544.4.001.57:681.3

MODELING OF VARIABLE HYDRAULIC CONDITIONS IN A HOT-HOUSE HEATING SYSTEM

[Synopsis of article by A. Z. Grishchenko, doctor of technical sciences, and G. N. Terlyayeva, engineer, pp 5-9]

[Text] The article describes a procedure for obtaining approximation models of hydraulic conditions in a two-circuit hot-house heating system with simultaneous use of a relaxation method for circuit consumption and an interference-proof algorithm for group calculation of arguments.

Compared to the traditional calculation of regulated heating networks and in comparison to simple approximation methods, the procedure allows significant reductions in time and economical use of memory in control mini- and microcomputers. Two illustrations, five references.

UDC 681.327

AN APPROACH TO SOLVING THE PROBLEM OF MICROPROGRAM MEMORY STRUCTURE OPTIMIZATION

[Synopsis of article by S. A. Shvedov, engineer, pp 9-12]

[Text] The article examines one of the possible approaches to solving the problem of microprogram memory optimization. Criteria are proposed for optimal distribution of microoperations in linear microprogram sections. Comparative evaluations of several microprogram memory structures are presented. Four references.

UDC 622.7:65.011.56

ADAPTIVE MODEL-BASED CALCULATION OF OPTIMAL WATER CONDITIONS FOR FIRST-STAGE MAGNETIC CONCENTRATION OF ORE IN MILLS

[Synopsis of article by A. P. Polishchuk, candidate of technical sciences, V. M. Davidchenko, V. N. Kobozev and V. G. Knyshev, engineers, pp 12-15]

[Text] The article sets forth a water flow optimization algorithm to obtain the maximum final classification in mill pourings with given ore consumption rates and classification parameters. The algorithm implements a searc'-free optimization method based on the use of an adaptive mathematical model whose parameters are identified by means of a modification of the Kachmakh method. Three references.

UDC 681.3:658.512.6:621.774.016

OPTIMIZATION OF CURRENT BRANCH PLANNING FOR THE PRODUCTION OF COLD-WORKED PIPE

[Synopsis of article by A. M. Vaynzof, Yu. A. Bannik, candidates of economic sciences, B. I. Kamen', candidate of technical sciences, and N. G. Mazhara, engineer, pp 15-17]

[Text] The article examines the questions of improving current branch planning for the production of cold-worked pipe on the basis of optimization calculations targeted toward increasing equipment utilization efficiency and reducing finished product hauling. A mathematical model of branch planning tasks is described, as are the final output document formats required for production planning.

UDC 65.014.12:519.72

SELECTION OF MONITORING INTERVALS IN ORGANIZATIONAL CONTROL SYSTEMS

[Synopsis of article by M. A. Guriyev, V. A. Toloknyanenko and A. G. Yaremenko, candidates of technical sciences, pp 18-20]

[Text] From the broadest informational standpoint the article examines the task of selecting monitoring intervals for managerial decision-making and establishes its relationship with the problem of establishing values for continuous time functions according to their selection. Five references.

UDC 658.5.011.56

DETERMINING REPUBLIC AUTOMATED CONTROL SYSTEM INFLUENCE ON REPUBLIC REVENUE

[Synopsis of article by M. T. Matveyev, doctor of economic sciences, V. M. Leshchenko and A. A. Chalyy, candidates of technical sciences, pp 21-22]

[Text] The article proposes a methodical approach to the determination of republic automated control system influence on the national economy based on the growth of pure production and the reduction of material consumption in republic economy production divisions in connection with the introduction of the republic automated control system.

UDC 658.011.012.56:681.3.06

PROBLEMS OF DEVELOPING INPUT LANGUAGES FOR CROSS-SYSTEM PROGRAM PREPARATION

[Synopsis of article by S. A. Anan'yevskiy, N. F. Trigub, candidates of technical sciences, and G. T. Todorashko, engineer, pp 23-25]

[Text] The article describes the problems and difficulties arising in the design of new algorithmic languages oriented toward the development of automated control systems for technological processes.

The article briefly covers the characteristics of three algorithmic languages, different in type and intent, used in the cross-system of program preparation. Four references.

UDC 681.3.068

DATA BASE OF A SYSTEM FOR AUTOMATED MICROPROGRAMMING OF MICROPROCESSOR DEVICES

[Synopsis of article by V. G. Kantor, engineer, pp 25-29]

[Text] The article describes the data base of a system for microprogramming automation, its design and its scope of application as the core of an operating system in one process of developing microprograms. A characteristic of the data base is its capacity for use in interactive debugging and rapid patching of microprograms at the symbolic level. Two illustrations, four references.

UDC 658.012.011.56

DEFINITION OF ECONOMIC EFFICIENCY INDICATORS FOR OPERATIONAL AUTOMATED CONTROL SYSTEMS

[Synopsis of article by A. A. Chalyy, candidate of technical sciences, and Ye. G. Gritsenko, candidate of economic sciences, pp 29-31]

[Text] As a supplement to current procedural documentation, the article proposes a procedure for determining the actual economic efficiency of an automated control system, on the basis of its stages of development. Two illustrations, two references.

UDC 620.169.1

SYSTEMS WITH ADAPTIVE PROPERTIES FOR THE CONTROL OF FULL-SCALE DURABILITY TESTING

[Synopsis of article by V. I. Litvak, candidate of technical sciences, pp 31-35]

[Text] The article examines the design principles and structural arrangements of several systems with adaptive properties which provide automatic support of maximum permissible loading speeds while securing precise reproduction characteristics for given loads and assuring system reliability in the face of changes in the process of testing specimen characteristics and external forces. Three illustrations, four references.

UDC 62.001.57:681.3

STRUCTURE MODELING FOR A PRODUCTION ASSOCIATION INFORMATION DISPLAY SYSTEM

[Synopsis of article by V. Ye. Khodakov, F. B. Rogal'skiy, candidates of technical sciences, and G. G. Savina, engineer, pp 35-38]

[Text] The article examines the questions involved in modeling the structure of a multiterminal information display system in a production association. The GPSS modeling language was used to construct an imitative model. Results of research on the parameters of an actual system are given. Three illustrations, 4 references.

UDC 681.32

EFFECTIVE MICROINSTRUCTION PIPELINE ORGANIZATION IN A HIGH-CAPACITY MICROPROCESSOR

[Synopsis of article by V. V. Novoselov, candidate of technical sciences, pp 38-41]

[Text] The article describes an adaptive microinstruction pipeline which achieves a throughput increase by reconfiguring the microprocessor structure when executing fragments of a microprogram control algorithm which vary in their characteristics. Three illustrations, two tables.

UDC 621.317.087:681.3

A METHOD FOR MACHINE DESIGN OF INITIAL TRANSDUCERS USING THE EXAMPLE OF LEVEL-GAUGE DESIGN

[Synopsis of article by Ye. A. Dokukin, engineer, pp 41-42]

[Text] A method is proposed for machine evaluation of optimal parameters for initial transducers used in level-gauges. This method assures the achievement of desired output characteristics for these devices. A calculation example is given. Two references.

UDC 681.3:658.381:622.2

AUTOMATION OF TIME SHEET ACCOUNTING FOR WORKERS IN MINE SURFACE FACILITIES

[Synopsis of article by V. N. Tereshchenko, T. N. Tereshchenko, V. F. Ukrainskiy, and N.A. Staroverov, engineers, pp 43-45]

[Text] The article indicates technical prerequisites for the implementation of an automated time sheet accounting system.

A variation of reading device use is described and the technical specifications of an inductive reading device, ISU(P), are given.

Information is provided on the results of testing experimental ISU(P) models. Two illustrations.

UDC 621.317.38

AUTOMATING THE CONTROL OF ELECTRICAL OPERATIONS IN DIRECT-HEATING FURNACES

[Synopsis of article by A. N. Tupikov, candidate of technical sciences, and Yu. M. Popovkin, engineer, pp 45-48]

[Text] The article introduces the calculation of electrical operation parameters used for automatic, programmed control of the thermal process in direct-heating furnaces. The general functional plan of an automated control system is examined. Three illustrations, three references.

UDC 621.311:621.316:683.3

HARDWARE RESOURCES AND PROGRAMMING FOR THE AUTOMATIC MONITORING AND CONTROL OF ELECTRICAL DEMAND

[Synopsis of article by V. D. Leporskiy, N. P. Lukash, candidates of technical sciences, A. S. Polgorodnik and Ye. L. Makovskiy, engineers, pp 48-50]

[Text] The article describes a set of technical resources for a typical radius system used to monitor and control electrical loads in rayon-level power grids. The article further gives information on the characteristics of such a system, its software characteristics and its prospects for further application. One illustration, three references.

UDC 681.325.53

INCREASING THE TIMING ACCURACY OF TIME/FREQUENCY TELEAUTOMATION SYSTEMS

[Synopsis of article by N. A. Chekhlatyy, engineer, and N. V. Zakharchenko, candidate of technical sciences, pp 51-54]

[Text] The article analyzes the effectiveness of two methods of shaping clock pulses and gives the results of research on the law of changing pulse dimension recognition. A functional diagram of a clock pulse driver is presented. Three illustrations, one table, five references.

UDC 681.3.06

REQUEST DISCIPLINE IN A MULTIPOINT LINK OF A DATA TELEPROCESSING SYSTEM

[Synopsis of article by Yu. F. Manerko, candidate of technical sciences, and V. A. Trukhnin, engineer, pp 54-56]

[Text] A modification of the cyclical query spatial discipline is proposed to increase the efficiency and operation of the multipoint links in a data teleprocessing system with single-terminal subscriber dialog points. Results of imitative modeling research on this discipline are given. Three illustrations, four references.

UDC 681.324

MINI- AND MICROCOMPUTER LOCAL CONTROL NETWORK ARCHITECTURE AND DESIGN PRINCIPLES

[Synopsis of article by V. A. Ivanov, candidate of technical sciences, A. A. Cherevko, engineer, and V. N. Yatsenko, engineer, pp 56-58]

[Text] The article examines the architecture of distributed control systems. The different data transmission system design variations for such a network are examined. The "Elektronika-60", SM4 computer local area network is described. Two illustrations, two references.

12746
CSO: 1863/66

- END -